From Uncertain to Safe: Conformal Adaptation of Diffusion Models for Safe PDE Control

Peyan Hu [*] [1] [2] **Xiaowei Qian** [*] [1] [3] **Wenhao Deng** [3] **Rui Wang** [1] [4] **Haodong Feng** [3] **Ruiqi Feng** [3] **Tao Zhang** [3]
Long Wei [3] **Yue Wang** [5] **Zhi-Ming Ma** [2] **Tailin Wu** [3]

Abstract

The application of deep learning for partial differential equation (PDE)-constrained control is gaining increasing attention. However, existing methods rarely consider safety requirements crucial in real-world applications. To address this limitation, we propose *Safe Diffusion Models for PDE Control* (SafeDiffCon), which introduce the uncertainty quantile as model uncertainty quantification to achieve optimal control under safety constraints through both post-training and inference phases. Firstly, our approach post-trains a pre-trained diffusion model to generate control sequences that better satisfy safety constraints while achieving improved control objectives via a reweighted diffusion loss, which incorporates the uncertainty quantile estimated using conformal prediction. Secondly, during inference, the diffusion model dynamically adjusts both its generation process and parameters through iterative guidance and fine-tuning, conditioned on control targets while simultaneously integrating the estimated uncertainty quantile. We evaluate SafeDiffCon on three control tasks: 1D Burgers' equation, 2D incompressible fluid, and controlled nuclear fusion problem. Results demonstrate that SafeDiffCon is the only method that satisfies all safety constraints, whereas other classical and deep learning baselines fail. Furthermore, while adhering to safety constraints, SafeDiffCon achieves the best control performance. The code can be found at https://github.com/AI4Science-WestlakeU/safediffcon.

1. Introduction

The control of physical systems described by partial differential equations (PDEs) is crucial in various scientific and engineering domains, including fluid dynamics (Hinze & Kunisch, 2001), controlled nuclear fusion (Schuster & Ariola, 2006), and mathematical finance (Soner, 2004). While traditional control algorithms have been extensively studied over the years (Li et al., 2006; Protas, 2008), advancements in neural networks have led to the development of numerous deep learning-based approaches (Farahmand et al., 2017; Holl et al., 2020; Hwang et al., 2022). Among these, diffusion models have gained increasing attention to PDE control due to their ability to model high-dimensional and nonlinear data (Vahdat et al., 2022; Li et al., 2024), and have demonstrated impressive performance in various control tasks (Chi et al., 2023; Wei et al., 2024; Hu et al., 2024).

However, existing deep learning-based methods for PDE control generally overlook *safety* – meaning ensuring control sequences satisfy predefined constraints, mitigating risks, and preventing hazards (Dawson et al., 2022; Liu et al., 2023a). In real-world applications, controlling PDE systems particularly requires addressing safety concerns (Barros et al., 1998; Argomedo et al., 2013). For instance, in fluid dynamics, minor errors can cause turbulence or structural damage, while in controlled nuclear fusion, violating safety constraints may lead to catastrophic failures. This lack of safety enforcement remains a major bottleneck in applying deep learning to scientific and engineering problems, posing a critical challenge for high-stakes applications.

Despite its importance, safe PDE control remains challenging. First, ensuring safety requires preventing methods without formal guarantees from interacting with the environment, restricting the approach to an offline setting with pre-collected data. However, such data is often suboptimal and includes unsafe samples, creating a significant gap between the observed distribution and the near-optimal, safe distribution (Xu et al., 2022; Liu et al., 2023a). Second, the method must resolve the inherent conflict between optimizing control performance and satisfying safety constraints (*e.g.*, aggressive control policies that reduce error may violate safety thresholds) (Liu et al., 2023a; Zheng et al., 2024).

[*]Equal contribution [1]Work done as an intern at Westlake University [2]Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China [3]School of Engineering, Westlake University, Hangzhou, China [4]Fudan University, Shanghai, China [5]Zhongguancun Academy, Beijing, China. Correspondence to: Tailin Wu <wutailin@westlake.edu.cn>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

To address these challenges, we propose *Safe Diffusion Models for PDE Control* (SafeDiffCon), a method that adapts diffusion models to varying safety constraints by quantifying their uncertainty. In offline settings, training data are often sub-optimal and unsafe, which causes the distribution learned by diffusion models to deviate significantly from the desired optimal and safe distribution. Inspired by conformal prediction (Vovk et al., 2005; Tibshirani et al., 2019), we propose quantifying diffusion models' safety uncertainty under distribution shifts and incorporating it through both post-training and inference phases. This uncertainty quantification, termed as *uncertainty quantile*, allows us to compute a *conformal interval*, within which the safety score of the diffusion model interacting with the real environment is expected to lie. By leveraging this uncertainty quantification for predicted safety, diffusion models can be optimized with a clear safety optimization objective, to satisfy various safety constraints by constraining their conformal intervals within safe boundaries. Specifically, we incorporate uncertainty quantification into both the post-training and inference-time fine-tuning processes. Technically, we perform *post-training with reweighted loss*, where the weighting accounts for the uncertainty quantile of safety instead of the original predicted safety score. This refinement improves the model's control performance while enhancing its adherence to safety constraints, leading to the generation of more optimal and safer control sequences. To further ensure the safety of PDE control, we introduce *inference-time fine-tuning*, where the diffusion model is iteratively adjusted with guidance and fine-tuning based on specific control targets and uncertainty quantile to improve the safety of the final outputs within few iterations.

Our main contributions are as follows: **(1)** We introduce safety constraints into deep learning-based control of PDE systems and propose our method SafeDiffCon. **(2)** To address significant distribution shifts, we quantify the safety uncertainty of diffusion models and design a conformal adaptation process, which incorporates post-training with a reweighted loss and inference-time fine-tuning to promote a safer and more optimal output distribution. **(3)** We design safe control tasks in the contexts of the 1D Burgers' equation, 2D incompressible fluid, and the controlled nuclear fusion scenario, and evaluate various methods. Extensive experiments on these datasets demonstrate that SafeDiffCon is the only method to successfully meet all safety constraints while achieving superior control objectives.

2. Related Work

2.1. Control of PDE Systems

The development of control methods in physical systems is critical across various scientific and engineering areas, including PID (Li et al., 2006), supervised learning (SL) (Holl et al., 2020; Hwang et al., 2022), reinforcement learning (RL) (Farahmand et al., 2017; Pan et al., 2018; Rabault et al., 2019), and physics-informed neural networks (PINNs) (Mowlavi & Nabi, 2023). Among these, PID is one of the earliest and most widely used method (Johnson & Moradi, 2005), known for its simplicity and effectiveness in regulating physical systems; however, it faces challenges in parameter tuning and struggles with highly nonlinear or time-varying systems. Model Predictive Control (MPC) (Garcia et al., 1989; Findeisen & Allgöwer, 2002; Schwenzer et al., 2021) is another well-known control strategy that optimizes a control sequence over a finite time horizon by solving an optimization problem at each time step, using a dynamic model of the system. As it requires solving an optimization problem in real-time at each time step, it may have a high computational cost and struggle to handle high-dimensional, complex systems effectively. Furthermore, the adjoint method (Protas, 2008) and PINNs (Mowlavi & Nabi, 2023) are also incorporated in PDE control, but they require an explicit form of the PDE. Currently, the diffusion model used in physical systems' control (Wei et al., 2024; Hu et al., 2024) integrates the learning of entire state trajectories and control sequences, enabling global optimization that incorporates the physical information learned by the model. However, it does not consider the important cases where safety constraints are required.

In the field of reinforcement learning (RL), several safe offline RL algorithms have emerged in recent years. CPQ (Xu et al., 2022) is the first practical safe offline RL method that assigns high costs to out-of-distribution (OOD) and unsafe actions and updates the value function as well as the policy only with safe actions. COptiDICE (Lee et al., 2022) is a DICE-based method and corrects the stationary distribution. CDT (Liu et al., 2023b) takes the decision transformer to solve safe offline RL problems as multi-objective optimization. More recent methods such as TREBI (Lin et al., 2023) and FISOR (Zheng et al., 2024) address this problem through diffusion model planning, leveraging the capability of diffusion models to model high-dimensional data. However, none of these algorithms take into account the uncertainty of the predicted safety from the perspective of conformal prediction, nor do they address PDE-constrained scenarios.

2.2. Conformal Prediction

Conformal prediction (Vovk et al., 2005) is a statistical framework that constructs prediction intervals guaranteed to contain the true label with a specified probability. Its validity could be compromised, however, by the violation of the core assumption of exchangeability due to distribution shifts in real-world scenarios (Chernozhukov et al., 2018; Hendrycks et al., 2018). Recent studies (Maxime Cauchois & Duchi, 2024) have extended conformal prediction to accommodate various distribution shifts. For example, Tibshirani et al. (2019) proposed weighted conformal prediction to handle covariate shift, where training and test data distributions differ. Podkopaev & Ramdas (2021) introduced reweighted conformal prediction and calibration techniques to address label shift using unlabeled target data. Adaptive conformal inference (Gibbs & Candes, 2021) provides valid prediction sets in online settings with unknown, time-varying distribution shifts without relying on exchangeability. Inspired by conformal prediction, we measure uncertainty based on the distribution of data generated by diffusion models for control problems. This is also the first time the perspective of conformal prediction has been applied to address safety control problems.

3. Preliminary

3.1. Problem Setup

We consider the following safe control problem of PDE-constrained systems:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \mathcal{J}(\mathbf{u},\mathbf{w}) \quad \text{s.t.} \quad \mathcal{C}(\mathbf{u},\mathbf{w}) = 0, \quad s(\mathbf{u}) \le s_0, \tag{1}$$

where $\mathbf{u}(t,\mathbf{x}) : [0,T] \times \Omega \mapsto \mathbb{R}^{d_\mathbf{u}}$ is the system's state trajectory with dimension $d_\mathbf{u}$ and $\mathbf{w}(t,\mathbf{x}) : [0,T] \times \Omega \mapsto \mathbb{R}^{d_\mathbf{w}}$ is the external control signal with dimension $d_\mathbf{w}$. They are both defined on the time range $[0,T] \subset \mathbb{R}$ and spatial domain $\Omega \subset \mathbb{R}^D$. $\mathcal{J}(\mathbf{u},\mathbf{w})$ is the objective of the control problem, and $\mathcal{C}(\mathbf{u},\mathbf{w}) = 0$ is the PDE constraint.

As for the safety constraint, $s(\mathbf{u})$ is the safety score and s_0 is the bound of the safety score. We need to minimize the control objective while satisfying PDE constraints and constraining the safety score to stay below the bound, which requires a careful balance between safety and control performance. Furthermore, it is important to note that safety and control performance are not on equal footing, and the pursuit of a better objective should be built upon ensuring safety (Knight, 2002; Cheng et al., 2019).

3.2. Diffusion Models and Diffusion Control

Diffusion models (Ho et al., 2020) learn data distribution from data in a generative way. They present impressive performance in a broad range of generation tasks. Diffusion models involve diffusion/denoising processes: the diffusion process $q(\mathbf{x}^{k+1}|\mathbf{x}^k) = \mathcal{N}(\mathbf{x}^{k+1}; \sqrt{\alpha_k}\mathbf{x}_k, (1-\alpha_k)\mathbf{I})$ corrupts the data distribution $p(\mathbf{x}_0)$ to a prior distribution $\mathcal{N}(\mathbf{0},\mathbf{I})$, and the denoising process $p_\theta(\mathbf{x}^{k-1}|\mathbf{x}^k) = \mathcal{N}(\mathbf{x}^{k-1}; \mu_\theta(\mathbf{x}^k, k), \sigma_k\mathbf{I})$ makes sampling in a reverse direction. Here k is the diffusion/denoising step, $\{\alpha_k\}_{k=1}^K$ and $\{\sigma_k\}_{k=1}^K$ are the noise and variance schedules. In practice, a denoising network $\boldsymbol{\epsilon}_\theta$ is trained to estimate the noise to be removed in each step. During inference, the iterative removal of $\boldsymbol{\epsilon}_\theta$ from the prior distribution could generate a new sample that follows the data distribution $p(\mathbf{x})$.

Recently, diffusion models (Wei et al., 2024; Hu et al., 2024) are applied to solve the control problem as in Eq. 1 without the safety constraint $s(\mathbf{u}) \le s_0$. For brevity, we only summarize the light version. These methods transform the physical constraint to a parameterized energy-based model (EBM) $E_\theta(\mathbf{u},\mathbf{w})$ with the correspondence $p(\mathbf{u},\mathbf{w}) \propto \exp(-E_\theta(\mathbf{u},\mathbf{w}))$. Then the problem is converted to an unconstrained optimization over $\mathbf{u}$ and $\mathbf{w}$ for all physical time steps simultaneously:

$$\mathbf{u}^*, \mathbf{w}^* = \arg\min_{\mathbf{u},\mathbf{w}} \left[E_\theta(\mathbf{u},\mathbf{w}) + \lambda \cdot \mathcal{J}(\mathbf{u},\mathbf{w}) \right], \tag{2}$$

where λ is a hyperparameter. To optimize E_θ, a denoising network $\boldsymbol{\epsilon}_\theta$ is trained to approximate $\nabla_{\mathbf{u},\mathbf{w}} E_\theta(\mathbf{u},\mathbf{w})$ by the following loss:

$$\begin{aligned}\mathcal{L}_{\text{diffusion}} = {} & \mathbb{E}_{k\sim U(1,K),(\mathbf{u},\mathbf{w})\sim p(\mathbf{u},\mathbf{w}),\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})} \\ & [\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_k}[\mathbf{u},\mathbf{w}] + \sqrt{1-\bar{\alpha}_k}\boldsymbol{\epsilon}, k)\|_2^2],\end{aligned} \tag{3}$$

where $\bar{\alpha}_k := \prod_{i=1}^k \alpha_i$. After $\boldsymbol{\epsilon}_\theta$ is trained, Eq. 2 can be optimized by sampling from an initial sample $(\mathbf{u}^K, \mathbf{w}^K) \sim \mathcal{N}(\mathbf{0},\mathbf{I})$, and iteratively running the following process[1]

$$\begin{aligned}(\mathbf{u}^{k-1}, \mathbf{w}^{k-1}) = {} & (\mathbf{u}^k, \mathbf{w}^k) - \eta_k \cdot \boldsymbol{\epsilon}_\theta([\mathbf{u}^k, \mathbf{w}^k], k) \\ & + \lambda_k \nabla_{\mathbf{u},\mathbf{w}} \mathcal{G}(\hat{\mathbf{u}}^k, \hat{\mathbf{w}}^k) + \xi, \quad \xi \sim \mathcal{N}\big(\mathbf{0}, \sigma_k^2\mathbf{I}\big)\end{aligned} \tag{4}$$

under the guidance of $\mathcal{G} = \mathcal{J}$ for $k = K, K-1, ..., 1$. Here $[\hat{\mathbf{u}}^k, \hat{\mathbf{w}}^k]$ is the noise-free estimation of $[\mathbf{u}^0, \mathbf{w}^0]$, $[\cdot,\cdot]$ denotes vector concatenation, and η_k and λ_k are denoising schedules. The final sampling step yields the solution $\mathbf{w}^0$ for the optimization problem in Eq. 2.

4. Method

Algorithm 1 Algorithm of SafeDiffCon

```
Require Training set $D_{\text{train}}$, Calibration set $D_{\text{cal}}$, control targets
$\theta \leftarrow$ Pre-train with $\mathcal{L}_{\text{diffusion}}$ on $D_{\text{train}}$
$\theta \leftarrow$ Post-train with $\mathcal{L}_{\text{post-train}}$ on $D_{\text{train}}$ and $D_{\text{cal}}$ (Sec 4.2, Alg 2)
$\theta, \mathbf{w} \leftarrow$ Fine-tune during guided inference with $\mathcal{G}$ and $\mathcal{L}_{\text{fine-tune}}$ on control targets and $D_{\text{cal}}$ (Sec 4.3, Alg 3)
return model parameters $\theta$ and control sequences $\mathbf{w}$
```

In this section, we introduce our proposed method SafeDiffCon, with its overall framework outlined in Figure 1 and Algorithm 1. Firstly, in Section 4.1, we propose the uncertainty quantile, which is the quantification of uncertainties. As shown in the middle of the figure, it is embedded

[1] In this paper, we use $\mathbf{u}$ to represent state trajectory across time, and $\mathbf{w}$ to represent control signal sequence across time. $\mathbf{u}^k$ denotes the full system trajectory across time at denoising step k.

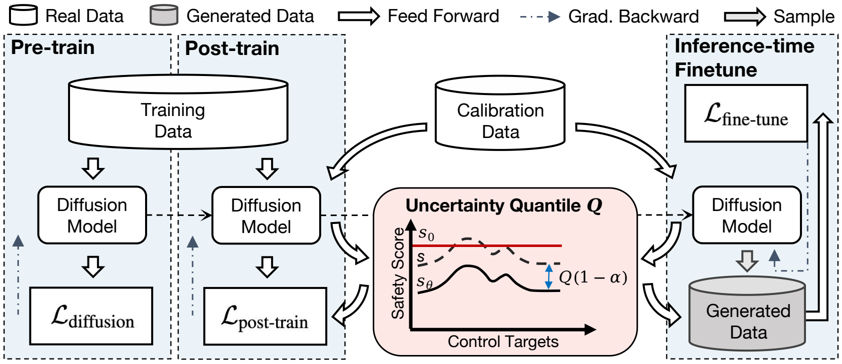

Figure 1: **Overview of SafeDiffCon**. First, we pre-train a diffusion model p_θ on the training data. Then, combined with the uncertainty quantile, we post-train the model to steer its distribution to safer regions with better control objectives. Finally, to improve performance and safety for specific control tasks, we conduct inference-time fine-tuning, again incorporating the uncertainty quantile into the process.

throughout the algorithm, originating from the concern that the model's predictive uncertainty may cause a gap between the actual and predicted safety scores, potentially leading to unsafe events not anticipated by the model.

Secondly, in Section 4.2 and 4.3, we introduce the post-training and inference-time fine-tuning based on the uncertainty quantile, respectively. In the post-training phase following pre-training, we employ a reweighted loss function to guide the model's output distribution to favor regions with more optimal objectives and greater safety in general, after which the post-trained model can subsequently be used for specific control tasks. As for inference-time fine-tuning, we aim to make task-specific adjustments based on the post-trained model, enabling better control performance and safety guarantee for the specific control tasks with few iterations.

4.1. Uncertainty Quantification of Diffusion Models

In offline safe control problems (Xu et al., 2022; Gu et al., 2022), the gap between pre-collected data and the target distribution exacerbates models' prediction errors, which can be critical in ensuring safety. To address this issue, we employ the conformal prediction technique to obtain uncertainty quantification for predicted safety, without requiring additional assumptions about the model and the data distribution.

The intuition behind the original conformal prediction (Vovk et al., 2005) is to use the trained model on the calibration set to obtain a score set. The score set is then used to endow the model's prediction with a *conformal (confidence) interval*. Under the assumption of exchangeability[2], the actual prediction during inference will lie within the conformal interval with a guaranteed probability. The details of conformal prediction are provided in Appendix D.

However, there is typically a distribution shift between the calibration set and the sequences generated by the diffusion model for control tasks during inference. Therefore, we account for this distribution shift, thereby introducing the *shifted score set*. Based on it, we get the uncertainty quantification metric named *uncertainty quantile* and finally the *conformal interval*. The detailed description of this process is as follows.

Calibration set and pre-training. To achieve the goal mentioned above, we first set aside a portion of the original training data as the *calibration set* D_{cal}, which will be used later to estimate the model's prediction errors. The remaining data, which will be applied to operations that can alter the model's outcomes, is referred to as the training data D_{train}. After pre-training with D_{train} as described in Eq. 3, we get the diffusion model p_θ which models the joint distribution of $[\mathbf{u}, \mathbf{w}]$.

Shifted score set under distribution shift. *Score set*, obtained on the calibration set, is a set of model prediction uncertainties regarding the safety score s, which is defined as

$$\mathcal{S} := \{|s(\mathbf{u}_\theta(\mathbf{w}_i)) - s(\mathbf{u}_i)| : (\mathbf{u}_i, \mathbf{w}_i) \in D_{\text{cal}}\} \cup \{\infty\}. \quad (5)$$

Here $\mathbf{u}_i$ and $\mathbf{w}_i$ are the system state trajectory and control signal sequence of the i^{th} example, $\mathbf{u}_\theta(\mathbf{w})$ is the system state trajectory conditioned on control $\mathbf{w}_i$ as predicted by the model, where θ is the model parameter.

Furthermore, taking into account the distribution shift between the calibration set D_{cal} and data generated based on control targets, we apply weighting to the score set $\mathcal{S}$ to get the *shifted score set* as

$$\tilde{\mathcal{S}} := \{\omega_{\text{norm}}(\mathbf{u}_i, \mathbf{w}_i)\Delta s_i : \Delta s_i \in \mathcal{S}\}. \quad (6)$$

Intuitively, this weighting is because each sample in the calibration set has a different probability of appearing in the

[2] Exchangeability means that exchanging examples in the calibration set and in inference does not alter their joint distribution.

final data distribution generated by the model. According to conformal prediction under covariate shift (Tibshirani et al., 2019), here the calculation of the weights is as follows:

$$\omega(\mathbf{u}_i, \mathbf{w}_i) := \frac{\tilde{p}(\mathbf{u}_i, \mathbf{w}_i)}{p(\mathbf{u}_i, \mathbf{w}_i)} \tag{7}$$

where $p(\mathbf{u}, \mathbf{w})$ is the probability density function of the calibration set and training set, and $\tilde{p}(\mathbf{u}, \mathbf{w})$ is the probability density function of the model-generated data during inference. ω_{norm} is the normalization of ω defined as

$$\omega_{\text{norm}}(\mathbf{u}_i, \mathbf{w}_i) = \frac{\omega(\mathbf{u}_i, \mathbf{w}_i)}{\sum_{(\mathbf{u}_i, \mathbf{w}_i) \in D_{\text{cal}}} \omega(\mathbf{u}_j, \mathbf{w}_j)}. \tag{8}$$

The specific calculation of ω_{norm} regarding SafeDiffCon will be detailed in Section 4.2.

Uncertainty quantile. Based on the shifted score set $\tilde{\mathcal{S}}$ (Eq. 6), the *uncertainty quantile* is Quantile$((1-\alpha)(1+\frac{1}{|D_{\text{cal}}|}); \tilde{\mathcal{S}})$[3], where $1-\alpha$ is the coverage probability and $|D_{\text{cal}}|$ is the cardinality of D_{cal}. The quantile represents the threshold below which a specific proportion of $\tilde{\mathcal{S}}$ falls. The adjustment term $1 + 1/|D_{\text{cal}}|$ accounts for the finite size of the calibration set, ensuring valid coverage even with limited data. Hereafter, it is abbreviated as $Q(1-\alpha; \tilde{\mathcal{S}})$.

We can then obtain the *conformal interval* as:

$$\begin{aligned}\text{CI}_\theta(1-\alpha, D_{\text{cal}}) := [&s(\mathbf{u}_\theta(\mathbf{w})) - Q(1-\alpha; \tilde{\mathcal{S}}),\\ &s(\mathbf{u}_\theta(\mathbf{w})) + Q(1-\alpha; \tilde{\mathcal{S}})].\end{aligned} \tag{9}$$

With at least $1-\alpha$ probability, the true s is covered by this interval, which can be demonstrated through the following theorem.

Theorem 4.1. *Assume that samples in the calibration set $D_{cal} \sim p$ are independent, and the test set $(\mathbf{u}, \mathbf{w}) \sim \tilde{p}$ is also independent with the calibration set. Assume p is absolutely continuous with respect to $\tilde{p}$, then*

$$\mathbb{P}(s(\mathbf{u}) \in CI_\theta(1-\alpha, D_{cal})) \geq 1 - \alpha. \tag{10}$$

(See Appendix A for proof.)

4.2. Post-training with Reweighted Loss

The aim of post-training is to transform the model's distribution into $p^*(\mathbf{u}, \mathbf{w}) \propto p(\mathbf{u}, \mathbf{w}) \cdot e^{-\mathcal{W}(\mathbf{u}, \mathbf{w})}$, where $\mathcal{W}(\mathbf{u}, \mathbf{w})$ contains both the control objective $\mathcal{J}$ and the safety constraints involving the model's uncertainty. Specifically, it is formulated as:

$$\mathcal{W}(\mathbf{u}, \mathbf{w}) = \max[s(\mathbf{u}) + Q(1-\alpha; \tilde{\mathcal{S}}) - s_0, 0] + \gamma \mathcal{J}(\mathbf{u}, \mathbf{w}), \tag{11}$$

where γ is the weight of $\mathcal{J}$.

So we propose the reweighted post-training loss as

$$\begin{aligned}&\mathcal{L}_{\text{post-train}} := \mathbb{E}_{k \sim U(1,K), (\mathbf{u}, \mathbf{w}) \sim p(\mathbf{u}, \mathbf{w}), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})}\\ &[e^{-\mathcal{W}(\mathbf{u}, \mathbf{w})} \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_k}[\mathbf{u}, \mathbf{w}] + \sqrt{1-\bar{\alpha}_k}\boldsymbol{\epsilon}, k)\|_2^2],\end{aligned} \tag{12}$$

which steers the generated data distribution towards high $e^{-\mathcal{W}}$ regions. Compared to the pre-training loss in Eq. 3, the samples are reweighted, assigning higher weights to those with lower $\mathcal{W}$. This loss function enables the model to adjust the generated data distribution to align with our target distribution, as described in the following theorem:

Theorem 4.2. *Trained with $\mathcal{L}_{post\text{-}train}$ in Eq. 12, the diffusion model's distribution is $p^*(\mathbf{u}, \mathbf{w}) \propto p(\mathbf{u}, \mathbf{w}) e^{-\mathcal{W}(\mathbf{u}, \mathbf{w})}$. (See Appendix A for proof.)*

Specifically, in the calculation of the uncertainty quantile Q, $\omega(\mathbf{u}_i, \mathbf{w}_i) = Cp_\theta(\mathbf{u}_i, \mathbf{w}_i) e^{-\mathcal{W}(\mathbf{u}_i, \mathbf{w}_i)} / p(\mathbf{u}_i, \mathbf{w}_i)$. In practical implementation, we let

$$\omega(\mathbf{u}_i, \mathbf{w}_i) = Ce^{-\mathcal{W}(\mathbf{u}_i, \mathbf{w}_i)}, \tag{13}$$

and the constant C will be eliminated during normalization. Through the following theorem, we demonstrate that when the KL divergence between p and p_θ is small, this approximation does not affect the results of the uncertainty quantile.

Theorem 4.3. *If $D_{KL}(p||p_\theta) \leq \epsilon$, then with $1 - C\sqrt{\epsilon}$ probability, Quantile$(1-\alpha; \sum_{i=1}^N w_{norm,i}\delta_{v_i})$ = Quantile$(1-\alpha; \sum_{i=1}^N \tilde{w}_{norm,i}\delta_{v_i})$, where $w_{norm,i}$ and $\tilde{w}_{norm,i}$ are the normalized w_i and $\tilde{w}_i$, $w_i = p_\theta(v_i)e^{-\mathcal{G}(v_i)}/p(v_i)$, and $\tilde{w}_i = e^{-\mathcal{G}(v_i)}$. (See Appendix A for proof.)*

The algorithm of post-training is provided in Algorithm 2 in Appendix E.2.

4.3. Inference-time Fine-tuning

During inference, knowing the specific control tasks, we guide the model to further optimize the generation of control sequences tailored to these tasks. We cyclically use the guidance to generate samples and then fine-tune the model parameters with these samples, thereby iteratively improving the model's safety and performance. In this process, similar to post-training, we incorporate the uncertainty quantile Q into both the guidance and fine-tuning loss to prevent violations of safety constraints caused by the model's uncertainty.

Guidance $\mathcal{G}$. Guidance is the first approach we adopt to steer the model's output toward satisfying both the control objectives and safety constraints. It plays a role during the sampling process of the diffusion model. The specific denoising step of implementing guidance follows Eq. 4, and the form of guidance still follows $\mathcal{W}$ in Eq. 11, *i.e.*

$$\mathcal{G}(\mathbf{u}, \mathbf{w}) = \mathcal{W}(\mathbf{u}, \mathbf{w}), \tag{14}$$

[3] It means obtaining the element with $(1-\alpha)(1+\frac{1}{|D_{\text{cal}}|})$'th quantile from the shifted score set $\tilde{\mathcal{S}}$.

incorporating the uncertainty quantile Q to account for safety risks arising from uncertainty.

Fine-tuning. The second approach for adjusting the output data distribution of the model is fine-tuning, which achieves the adjustment by optimizing the model parameters θ. Specifically, retaining the computation graph for all denoising steps with respect to θ would result in an unmanageable memory overhead. Therefore, when we need to keep the computation graph during denoising for gradient calculation, we only retain the computation graph of the final denoising step. To optimize the safety score and objective simultaneously, we form the fine-tune loss as:

$$\mathcal{L}_{\text{fine-tune}} = \sum_{(\mathbf{u}_\theta, \mathbf{w}_\theta) \in D_{\text{sampled}}} \mathcal{W}(\mathbf{u}_\theta, \mathbf{w}_\theta), \tag{15}$$

where $\mathbf{u}_\theta$ and $\mathbf{w}_\theta$ are from D_{sampled} sampled according to the guidance described above.

The entire algorithm of inference-time finetuning is provided in Algorithm 3 in Appendix E.3.

5. Experiment

We design three safe PDE control scenarios, including the 1D Burgers' equation, 2D incompressible fluid, and controlled nuclear fusion problem. These experiments aim to address the following three questions: (1) Does introducing the uncertainty quantile enable SafeDiffCon to meet safety requirements? (2) Can SafeDiffCon achieve superior control performance under safety constraints? (3) Do all components of our proposed algorithm contribute effectively to its performance?

For comparison, we choose imitation learning method Behavior Cloning (*BC*) (Pomerleau, 1988), and safe reinforcement learning and imitation learning methods involving *BC* with safe data filtering (*BC-Safe*), Constrained Decision Transformer (*CDT*) (Liu et al., 2023b) and diffusion-based method *TREBI* (Lin et al., 2023). Note that *CDT* shows the best performance in the offline Safe RL benchmark OSRL (Liu et al., 2023a). In addition, we combine the physical system control method Supervised Learning (Hwang et al., 2022) and Model Predictive Control (Schwenzer et al., 2021) with the Lagrangian approach (Chow et al., 2018) (*SL-Lag, MPC-Lag*) to enforce safety constraints. We also apply the classical control method *PID* (Li et al., 2006). We provide the code here.

5.1. 1D Burgers' Equation

Experiment settings. 1D Burgers' equation is a fundamental equation that governs various physical systems including fluid dynamics and gas dynamics. Here we follow previous works (Hwang et al., 2022; Mowlavi & Nabi, 2023) and consider the Dirichlet boundary condition along with an external force $\mathbf{w}(t, x)$.

Table 1: **Results of 1D Burgers' equation.** Gray: there are unsafe trajectories. Black: all trajectories are safe. **Bold**: safe trajectories with *lowest* $\mathcal{J}$.

Methods	$\mathcal{J} \downarrow$	$\mathcal{R}_{\text{sample}} \downarrow$	$\mathcal{R}_{\text{time}} \downarrow$	$\mathcal{R}_{\text{point}} \downarrow$
BC	0.0001	38%	13%	1.2%
BC-Safe	0.0002	14%	3%	0.2%
PID	0.0968	0%	0%	0.0%
SL-Lag	0.0115	0%	0%	0.0%
MPC-Lag	0.0092	0%	0%	0.0%
CDT	0.0012	16%	3%	0.2%
TREBI	0.0074	0%	0%	0.0%
SafeDiffCon	**0.0016**	0%	0%	0.0%

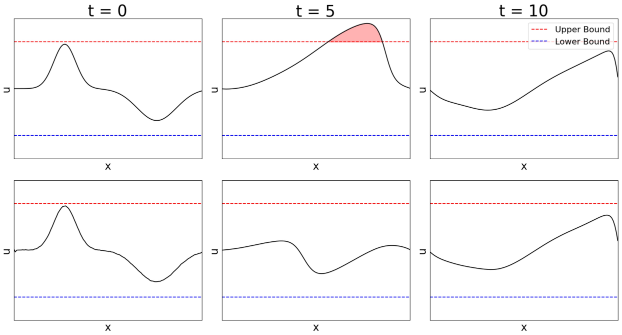

Figure 2: **Visualizations of the 1D Burgers' equation.** The top row shows the original trajectory corresponding to the control target, and the bottom row is the trajectory controlled by SafeDiffCon.

Given a target state $\mathbf{u}_d(x)$, the control objective $\mathcal{J}$ is to minimize the control error between the final state $\mathbf{u}_T$ and the target state $\mathbf{u}_d$.

$$\mathcal{J} := \int_\Omega |\mathbf{u}(T, x) - \mathbf{u}_d(x)|^2 \mathrm{d}x. \tag{16}$$

The safety score is defined as:

$$s(\mathbf{u}) := \sup_{(t,x) \in [0,T] \times \Omega} \{\mathbf{u}(t, x)^2\}, \tag{17}$$

with the safety constraint s_0. If $s(\mathbf{u}) > s_0$, the state trajectory $\mathbf{u}$ is unsafe, otherwise, it is safe. We compute three unsafe rates to assess the safety levels of different methods' control results. $\mathcal{R}_{\text{sample}}$ denotes the proportion of unsafe trajectories among total trajectories[4]; $\mathcal{R}_{\text{time}}$ denotes the proportion of unsafe timesteps among all timesteps; $\mathcal{R}_{\text{point}}$ denotes the proportion of unsafe spatial lattice points in all spatial lattice points across all time steps. More details can be found in Appendix F.1.

[4] If any point in the full trajectory is unsafe, this trajectory is unsafe. So $\mathcal{R}_{\text{sample}}$ is the most stringent metric.

Results. In Table 1, We report the results of the control objective $\mathcal{J}$ and safety metrics of different methods. SafeDiffCon can meet the safety constraint and achieve the best control objective at the same time. As shown in Figure 2, given the initial condition and the final state (control target), SafeDiffCon can control a state trajectory that satisfies the safety constraint and control target. Other methods either suffer from constraint violations or suboptimal objectives. *BC* and *BC-Safe* trained from expert trajectories fail to meet the safety constraints, showing that simple behavior cloning is not feasible under the safety constraints. *SL-Lag* and *MPC-Lag* attempt to use the Lagrangian method to balance the control objective and safety, but this coupled training program makes it difficult to find the right balance, with poor control performance. *CDT* uses the complex Transformer architecture, which can achieve low control error, but it can not meet safety constraints. The diffusion-based method *TREBI* sacrifices too much control performance to satisfy the safety constraints, because its error bound is soft, and the safety constraints can be easily violated.

5.2. 2D Incompressible Fluid

Experiment settings. We then consider the control problems of the 2D incompressible fluid, which follows the Navier-Stokes equation. Following previous works (Holl et al., 2020; Wei et al., 2024; Hu et al., 2024), the control task we consider is to maximize the amount of smoke that passes through the target bucket in the fluid flow with obstacles and openings, while constraining the amount of smoke passing through the dangerous region under the safety bound. Specifically, referring to Figure 3, the control objective $\mathcal{J}$ is defined as the negative ratio of smoke passing through the target bucket located at the center top, while the safety score s corresponds to the ratio of smoke entering the hazardous red region. It is important to note that there is a trade-off between controlling the flow through the hazardous region and achieving a more optimal control objective, which imposes higher demands on the algorithm. We set the safety score bound as $s_0 = 0.1$.

Moreover, this control task is particularly challenging due to its specific setup: not only does it require indirect control, which means that control can only be applied to the peripheral region, but the spatial control parameters also reach as many as 1,792. As for safety, among all the training data, 53.1% of the samples are unsafe, meaning their safety score s exceeds the bound $s_0 = 0.1$. The average safety score of the dataset is 0.3215. Other details can be found in Appendix G.1.

Results. We report results of SafeDiffCon and baselines in Table 2. Here *SL-Lag* and *MPC-Lag* fail to achieve reasonable control results because of the complex dynamics, and we define $\mathcal{R}$ as the rate of unsafe samples. Additionally, we introduce another metric called Safety Violation Magnitude $SVM = \max[s - s_0, 0]$. When s does not exceed the bound s_0, this metric is 0. If s exceeds s_0, the metric reflects the amount by which it is surpassed. From the results, it is clear that our method is the only one to successfully satisfy the safety constraint, even surpassing methods that only consider safety like BC-Safe, which demonstrates the effectiveness of post-training and fine-tuning with the uncertainty quantile.

Table 2: **2D incompressible fluid control results.** Gray: there are unsafe trajectories. Black: all trajectories are safe. **Bold**: safe trajectories with *lowest* $\mathcal{J}$.

Methods	$\mathcal{J} \downarrow$	$SVM \downarrow$	$\mathcal{R} \downarrow$
BC	-0.7104	0.7156	88%
BC-Safe	-0.2520	0.0330	8%
CDT	-0.7025	0.2519	30%
TREBI	-0.7019	0.0808	18%
SafeDiffCon	**-0.3548**	0.0000	0%

5.3. Tokamak Fusion Reactor

Experiment settings. A tokamak is a device designed to achieve controlled nuclear fusion by confining high-temperature plasma within a toroidal magnetic field, enabling conditions suitable for fusion reactions (Federici et al., 2001). The realization of controlled fusion could provide humanity with a clean, safe, and nearly limitless energy source, free from greenhouse gases, nuclear accident risks, and fuel supply imbalances (Schuster & Ariola, 2006). Safety in a tokamak is crucial to prevent plasma instabilities and disruptions (Kates-Harbeck et al., 2019), which can damage the device worth billions of dollars.

Controlling the plasma state in a tokamak is critical for maintaining stability and optimizing fusion performance, which often requires precise management of parameters like beta (β_p), safety factor (q_{95}), and internal inductance (l_i). Based on the previous works (Seo et al., 2021; 2022), we set the control objective to make β_p and l_i reach specific target values β_p^* and l_i^* at each time step, while the safety constraint requires q_{95} to remain above the defined safety bound at all times. In this case, the objective $\mathcal{J}$ and safety score s are formulated as follows:

$$\mathcal{J} := \int_{\Omega \times [0,T]} (|\beta_p(t,x) - \beta_p^*(x)|^2 \tag{18}$$

$$+ |l_i(t,x) - l_i^*(x)|^2) \mathrm{d}x\mathrm{d}t, \tag{19}$$

$$s := - \inf_{(t,x) \in [0,T] \times \Omega} \{q_{95}(t,x)\}. \tag{20}$$

And the safety score bound is $s_0 = -4.98$. We use the same initial state and tokamak simulation environment as in the previous work (Seo et al., 2022), setting time-varying

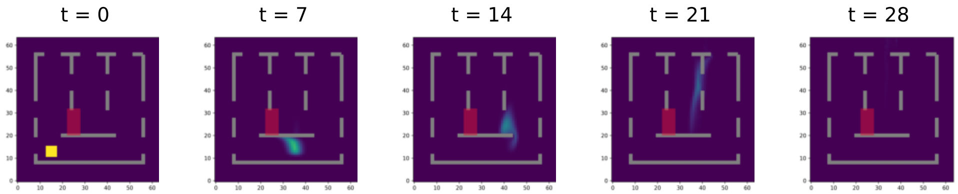

Figure 3: **Visualization of the 2D incompressible fluid control by our SafeDiffCon.** By controlling the fluid on the outside margin, the yellow smoke is successfully maneuvered to the center top exit while avoiding the red unsafe region.

random targets within the reasonable ranges proposed in that study. The control sequences for training are collected using the reinforcement learning model trained in the previous work (Seo et al., 2022), where 71.18% of the samples in the training dataset are unsafe.

Table 3: **Results of the tokamak fusion reactor.** Gray: there are unsafe trajectories. Black: all trajectories are safe. **Bold**: safe trajectories with *lowest* $\mathcal{J}$.

Methods	$\mathcal{J} \downarrow$	$\mathcal{R}_{\text{sample}} \downarrow$	$\mathcal{R}_{\text{time}} \downarrow$
BC	0.0610	42%	1.34%
BC-Safe	0.0811	4%	0.03%
SL-Lag	0.8812	0%	0.00%
MPC-Lag	0.8659	0%	0.00%
CDT	0.0071	8%	0.54%
TREBI	0.0261	0%	0.00%
SafeDiffCon	**0.0121**	0%	0.00%

Results. We provide the results of control and safety metrics in Table 3, where the definition of $\mathcal{R}_{\text{sample}}$ and $\mathcal{R}_{\text{time}}$ are the same as Section 5.1. Due to the differences in the input and output dimensions of the controller and the lack of clarity regarding the device's specific details, PID cannot be directly applied here. Results show that SafeDiffCon successfully achieves full compliance with safety constraints. Moreover, among all algorithms that satisfy safety constraints, our algorithm achieves the lowest $\mathcal{J}$, indicating that it also achieves the best control performance. These results verify our statement that our algorithm is not only safe but also effective.

5.4. Ablation Study

To demonstrate the necessity and effectiveness of each component of our algorithm, we conduct three ablation experiments on the 2D incompressible fluid control problem by removing post-training, inference-time fine-tuning, and the uncertainty quantile Q, respectively.

It can be observed that post-training and fine-tuning, as two methods that influence the model's output distribution, are both essential for meeting safety constraints. Removing either one results in the algorithm failing to achieve safety, highlighting the indispensable roles of these two components. Furthermore, when the uncertainty quantile is removed, which means Q is set to 0, the algorithm also fails to meet safety requirements. This demonstrates that without considering the model's uncertainty, its safety predictions are inaccurate, leading to the occurrence of unsafe events.

Table 4: **Results of the ablation study.** We compare SafeDiffCon with SafeDiffCon w/o post-training, inference-time fine-tuning and the uncertainty quantile Q.

Methods	$\mathcal{J} \downarrow$	$SVM \downarrow$	$\mathcal{R} \downarrow$
SafeDiffCon	-0.3548	0.0000	0%
w/o post-training	-0.4911	0.0849	12%
w/o fine-tuning	-0.4692	0.0247	8%
w/o Q	-0.6364	0.0623	12%

6. Limitation and Future Work

Firstly, our method is not constrained by specific scenarios, meaning it can be applied to more real-world tasks, which is our future work. Secondly, we consider extending this method to other generative methods that require constraints, not just the diffusion model. Finally, in the future, we will explore the possibility of developing a stricter coverage of the safety score that sacrifices less accuracy while still ensuring safety.

7. Conclusion

In this paper, we have introduced *Safe Diffusion Models for PDE Control* (SafeDiffCon), a novel approach that integrates safety constraints into deep learning-based control of PDE systems. By quantifying safety uncertainty using conformal prediction and introducing a reweighted loss during post-training, our method effectively handles distribution shifts and ensures safety while maintaining competitive control performance. Furthermore, inference-time fine-tuning allows dynamic adjustment of the model to meet specific control targets. The results of experiments on diverse control tasks, including 1D Burgers' equation, 2D incompressible fluid, and controlled nuclear fusion, demonstrate that SafeDiffCon reliably satisfies safety constraints and outperforms existing methods. This work paves the way for safer deployment of machine learning techniques in high-stakes real-world applications.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. Furthermore, our work makes Machine Learning applications safer in real-world scenarios. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

Acknowledgements

We thank Yuchen Yang for the discussions and for providing feedback on our manuscript. We gratefully acknowledge the support of Westlake University Research Center for Industries of the Future and Westlake University Center for High-performance Computing. The content is solely the responsibility of the authors and does not necessarily represent the official views of the funding entities.

References

Argomedo, F. B., Witrant, E., Prieur, C., Bremond, S., Nouailletas, R., and Artaud, J.-F. Lyapunov-based distributed control of the safety-factor profile in a tokamak plasma. *Nuclear Fusion*, 53(3):033005, 2013.

Barros, E., des Santos, and M.V.D. A safe, accurate intravenous infusion control system. *IEEE Micro*, 18(5): 12–21, 1998. doi: 10.1109/40.735940.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

Cheng, R., Orosz, G., Murray, R. M., and Burdick, J. W. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 3387–3395, 2019.

Chernozhukov, V., Wüthrich, K., and Yinchu, Z. Exact and robust conformal inference methods for predictive machine learning with dependent data. In *Conference On learning theory*, pp. 732–749. PMLR, 2018.

Chi, C., Xu, Z., Feng, S., Cousineau, E., Du, Y., Burchfiel, B., Tedrake, R., and Song, S. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, pp. 02783649241273668, 2023.

Chow, Y., Ghavamzadeh, M., Janson, L., and Pavone, M. Risk-constrained reinforcement learning with percentile risk criteria. *Journal of Machine Learning Research*, 18 (167):1–51, 2018.

Chung, H., Kim, J., Mccann, M. T., Klasky, M. L., and Ye, J. C. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=OnD9zGAGT0k.

Dawson, C., Qin, Z., Gao, S., and Fan, C. Safe nonlinear control using robust neural lyapunov-barrier functions. In *Conference on Robot Learning*, pp. 1724–1735. PMLR, 2022.

Dhariwal, P. and Nichol, A. Diffusion Models Beat GANs on Image Synthesis, June 2021. URL http://arxiv.org/abs/2105.05233. arXiv:2105.05233 [cs, stat].

Farahmand, A.-m., Nabi, S., and Nikovski, D. N. Deep reinforcement learning for partial differential equation control. In *2017 American Control Conference (ACC)*, pp. 3120–3127, 2017. doi: 10.23919/ACC.2017.7963427.

Federici, G., Skinner, C. H., Brooks, J. N., Coad, J. P., Grisolia, C., Haasz, A. A., Hassanein, A., Philipps, V., Pitcher, C. S., Roth, J., et al. Plasma-material interactions in current tokamaks and their implications for next step fusion reactors. *Nuclear Fusion*, 41(12):1967, 2001.

Findeisen, R. and Allgöwer, F. An introduction to nonlinear model predictive control. In *21st Benelux meeting on systems and control*, volume 11, pp. 119–141. Veldhoven, 2002.

Garcia, C. E., Prett, D. M., and Morari, M. Model predictive control: Theory and practice—a survey. *Automatica*, 25 (3):335–348, 1989.

Gibbs, I. and Candes, E. Adaptive conformal inference under distribution shift. *Advances in Neural Information Processing Systems*, 34:1660–1672, 2021.

Gu, S., Yang, L., Du, Y., Chen, G., Walter, F., Wang, J., and Knoll, A. A review of safe reinforcement learning: Methods, theory and applications. *arXiv preprint arXiv:2205.10330*, 2022.

Hendrycks, D., Mazeika, M., Wilson, D., and Gimpel, K. Using trusted data to train deep networks on labels corrupted by severe noise. *Advances in neural information processing systems*, 31, 2018.

Hinze, M. and Kunisch, K. Second order methods for optimal control of time-dependent fluid flow. *SIAM Journal on Control and Optimization*, 40(3):925–946, 2001.

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

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

Holl, P., Thuerey, N., and Koltun, V. Learning to control pdes with differentiable physics. In *International Conference on Learning Representations*, 2020.

Hu, P., Wang, R., Zheng, X., Zhang, T., Feng, H., Feng, R., Wei, L., Wang, Y., Ma, Z.-M., and Wu, T. Wavelet diffusion neural operator. *arXiv preprint arXiv:2412.04833*, 2024.

Hwang, R., Lee, J. Y., Shin, J. Y., and Hwang, H. J. Solving pde-constrained control problems using operator learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 4504–4512, 2022.

Janner, M., Du, Y., Tenenbaum, J., and Levine, S. Planning with diffusion for flexible behavior synthesis. *Proceedings of Machine Learning Research*, 162:9902–9915, 17–23 Jul 2022.

Johnson, M. A. and Moradi, M. H. *PID control*. Springer, 2005.

Kates-Harbeck, J., Svyatkovskiy, A., and Tang, W. Predicting disruptive instabilities in controlled fusion plasmas through deep learning. *Nature*, 568(7753):526–531, 2019.

Knight, J. C. Safety critical systems: challenges and directions. In *Proceedings of the 24th international conference on software engineering*, pp. 547–550, 2002.

Lee, J., Paduraru, C., Mankowitz, D. J., Heess, N., Precup, D., Kim, K.-E., and Guez, A. COptiDICE: Offline constrained reinforcement learning via stationary distribution correction estimation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=FLA55mBee6Q.

Lei, J., G'Sell, M. G., Rinaldo, A., Tibshirani, R. J., and Wasserman, L. A. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113:1094 – 1111, 2016. URL https://api.semanticscholar.org/CorpusID:13741419.

Li, T., Biferale, L., Bonaccorso, F., Scarpolini, M. A., and Buzzicotti, M. Synthetic lagrangian turbulence by generative diffusion models. *Nature Machine Intelligence*, pp. 1–11, 2024.

Li, Y., Ang, K. H., and Chong, G. Pid control system analysis and design. *IEEE Control Systems Magazine*, 26 (1):32–41, 2006. doi: 10.1109/MCS.2006.1580152.

Lin, Q., Tang, B., Wu, Z., Yu, C., Mao, S., Xie, Q., Wang, X., and Wang, D. Safe offline reinforcement learning with real-time budget constraints. In *International Conference on Machine Learning*, pp. 21127–21152. PMLR, 2023.

Liu, Z., Guo, Z., Lin, H., Yao, Y., Zhu, J., Cen, Z., Hu, H., Yu, W., Zhang, T., Tan, J., et al. Datasets and benchmarks for offline safe reinforcement learning. *arXiv preprint arXiv:2306.09303*, 2023a.

Liu, Z., Guo, Z., Yao, Y., Cen, Z., Yu, W., Zhang, T., and Zhao, D. Constrained decision transformer for offline safe reinforcement learning. In *International Conference on Machine Learning*, pp. 21611–21630. PMLR, 2023b.

Maxime Cauchois, Suyash Gupta, A. A. and Duchi, J. C. Robust validation: Confident predictions even when distributions shift. *Journal of the American Statistical Association*, 0(0):1–66, 2024. doi: 10.1080/01621459.2023.2298037. URL https://doi.org/10.1080/01621459.2023.2298037.

Mowlavi, S. and Nabi, S. Optimal control of pdes using physics-informed neural networks. *Journal of Computational Physics*, 473:111731, 2023.

Pan, Y., Farahmand, A.-m., White, M., Nabi, S., Grover, P., and Nikovski, D. Reinforcement learning with function-valued action spaces for partial differential equation control. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 3986–3995. PMLR, 10–15 Jul 2018.

Podkopaev, A. and Ramdas, A. Distribution-free uncertainty quantification for classification under label shift. In de Campos, C. and Maathuis, M. H. (eds.), *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pp. 844–853. PMLR, 27–30 Jul 2021. URL https://proceedings.mlr.press/v161/podkopaev21a.html.

Pomerleau, D. A. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

Protas, B. Adjoint-based optimization of pde systems with alternative gradients. *Journal of Computational Physics*, 227(13):6490–6510, 2008.

Rabault, J., Kuchta, M., Jensen, A., Réglade, U., and Cerardi, N. Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control. *Journal of fluid mechanics*, 865:281–302, 2019.

Schuster, E. and Ariola, M. The role of controls in nuclear fusion. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 2263–2270. IEEE, 2006.

Schwenzer, M., Ay, M., Bergs, T., and Abel, D. Review on model predictive control: An engineering perspective. *The International Journal of Advanced Manufacturing Technology*, 117(5):1327–1349, 2021.

Seo, J., Na, Y.-S., Kim, B., Lee, C., Park, M., Park, S., and Lee, Y. Feedforward beta control in the kstar tokamak by deep reinforcement learning. *Nuclear Fusion*, 61(10): 106010, 2021.

Seo, J., Na, Y.-S., Kim, B., Lee, C., Park, M., Park, S., and Lee, Y. Development of an operation trajectory design algorithm for control of multiple 0d parameters using deep reinforcement learning in kstar. *Nuclear Fusion*, 62 (8):086049, 2022.

Soner, H. M. *Stochastic optimal control in finance*. Scuola normale superiore, 2004.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

Tibshirani, R. J., Barber, R. F., Candès, E. J., and Ramdas, A. Conformal prediction under covariate shift. In *Neural Information Processing Systems*, 2019. URL https://api.semanticscholar.org/CorpusID:115140768.

Vahdat, A., Williams, F., Gojcic, Z., Litany, O., Fidler, S., Kreis, K., et al. Lion: Latent point diffusion models for 3d shape generation. *Advances in Neural Information Processing Systems*, 35:10021–10039, 2022.

Vovk, V., Gammerman, A., and Shafer, G. *Algorithmic learning in a random world*, volume 29. Springer, 2005.

Wei, L., Hu, P., Feng, R., Feng, H., Du, Y., Zhang, T., Wang, R., Wang, Y., Ma, Z.-M., and Wu, T. A generative approach to control complex physical systems. *Advances in Neural Information Processing Systems*, 2024.

Xu, H., Zhan, X., and Zhu, X. Constraints penalized q-learning for safe offline reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8753–8760, 2022.

Zheng, Y., Li, J., Yu, D., Yang, Y., Li, S. E., Zhan, X., and Liu, J. Safe offline reinforcement learning with feasibility-guided diffusion model. *arXiv preprint arXiv:2401.10700*, 2024.

A. Theory

In this section, we provide the proof of the theorems mentioned in the main text.

Theorem A.1. *Assume samples in the calibration set $D_{cal} \sim p$ are independent, and the test set $(\mathbf{u}, \mathbf{w}) \sim \tilde{p}$ is also independent with the calibration set. Assume p is absolutely continuous with respect to $\tilde{p}$, then*

$$\mathbb{P}(s(\mathbf{u}) \in CI_\theta(1-\alpha, D_{cal})) \geq 1-\alpha. \tag{21}$$

Proof. According to Lemma 2 in the previous work (Tibshirani et al., 2019), the calibration set and test set are weighted exchangeable (defined in Definition 1 in the previous work (Tibshirani et al., 2019)), with the weight function ω.

According to Lemma 3 in the previous work (Tibshirani et al., 2019), let the $(\mathbf{u}, \mathbf{w})$ pairs be the random variables, we can obtain that

$$\mathbb{P}(s(\mathbf{u}) \leq s(\mathbf{u}_\theta(\mathbf{w})) + Q(1-\alpha; \tilde{\mathcal{S}})) \geq 1-\alpha. \tag{22}$$

Similarly, it can be derived that

$$\mathbb{P}(s(\mathbf{u}_\theta(\mathbf{w})) - Q(1-\alpha; \tilde{\mathcal{S}}) \leq s(\mathbf{u})) \geq 1-\alpha. \tag{23}$$

Thus

$$\mathbb{P}(s(\mathbf{u}) \in \mathrm{CI}_\theta(1-\alpha, D_{\mathrm{cal}})) \geq 1-\alpha. \tag{24}$$

□

Theorem A.2. *Trained with $\mathcal{L}_{post\text{-}train}$*

$$\mathcal{L}_{post\text{-}train} = \mathbb{E}_{k\sim U(1,K),(\mathbf{u},\mathbf{w})\sim p(\mathbf{u},\mathbf{w}),\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}[e^{-\mathcal{W}(\mathbf{u},\mathbf{w})}\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_k}[\mathbf{u},\mathbf{w}] + \sqrt{1-\bar{\alpha}_k}\boldsymbol{\epsilon}, k)\|_2^2], \tag{25}$$

the diffusion model's distribution is $p^*(\mathbf{u}, \mathbf{w}) \propto p(\mathbf{u}, \mathbf{w})e^{-\mathcal{W}(\mathbf{u},\mathbf{w})}$.

Proof. Trained with the diffusion loss

$$\mathcal{L}_{\text{diffusion}} = \mathbb{E}_{k\sim U(1,K),(\mathbf{u},\mathbf{w})\sim p(\mathbf{u},\mathbf{w}),\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_k}[\mathbf{u},\mathbf{w}] + \sqrt{1-\bar{\alpha}_k}\boldsymbol{\epsilon}, k)\|_2^2], \tag{26}$$

the diffusion model's distribution is p, thus to learn the distribution p^*, the loss should be

$$\begin{aligned}
\mathcal{L} &= \mathbb{E}_{k\sim U(1,K),(\mathbf{u},\mathbf{w})\sim p^*(\mathbf{u},\mathbf{w}),\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_k}[\mathbf{u},\mathbf{w}] + \sqrt{1-\bar{\alpha}_k}\boldsymbol{\epsilon}, k)\|_2^2] && (27)\\
&= \mathbb{E}_{k\sim U(1,K),\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}[\int \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_k}[\mathbf{u},\mathbf{w}] + \sqrt{1-\bar{\alpha}_k}\boldsymbol{\epsilon}, k)\|_2^2 \cdot p^*(\mathbf{u},\mathbf{w})\mathbf{dudw}] && (28)\\
&\propto \mathbb{E}_{k\sim U(1,K),\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}[\int \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_k}[\mathbf{u},\mathbf{w}] + \sqrt{1-\bar{\alpha}_k}\boldsymbol{\epsilon}, k)\|_2^2 \cdot p(\mathbf{u},\mathbf{w})e^{-\mathcal{W}(\mathbf{u},\mathbf{w})}\mathbf{dudw}] && (29)\\
&= \mathbb{E}_{k\sim U(1,K),(\mathbf{u},\mathbf{w})\sim p(\mathbf{u},\mathbf{w}),\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}[e^{-\mathcal{W}(\mathbf{u},\mathbf{w})}\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_k}[\mathbf{u},\mathbf{w}] + \sqrt{1-\bar{\alpha}_k}\boldsymbol{\epsilon}, k)\|_2^2]. && (30)
\end{aligned}$$

□

Theorem A.3. *If the KL divergence of the real distribution p and the distribution p_θ learned by the model is less than ϵ, then with $1 - C\sqrt{\epsilon}$ probability, the quantile $Quantile(1-\alpha; \sum_{i=1}^N w_{norm,i}\delta_{v_i}) = Quantile(1-\alpha; \sum_{i=1}^N \tilde{w}_{norm,i}\delta_{v_i})$, where $w_{norm,i}$ and $\tilde{w}_{norm,i}$ are the normalized w_i and $\tilde{w}_i$, $w_i = \frac{p_\theta(v_i)e^{-\mathcal{W}(v_i)}}{p(v_i)}$ and $\tilde{w}_i = e^{-\mathcal{W}(v_i)}$.*

Proof. Since $D_{KL}(p||p_\theta) = \int p(x)\log\frac{p(x)}{p_\theta(x)}dx \le \epsilon$, we can get

$$\mathbb{P}(\mathcal{A} = \{|p(x) - p_\theta(x)| > \delta\}) \le \frac{\int_{\mathcal{A}} |p(x) - p_\theta(x)|dx}{\delta} \tag{31}$$

$$\le \frac{\int_{\Omega} |p(x) - p_\theta(x)|dx}{\delta} \tag{32}$$

$$= \frac{2TV(p, p_\theta)}{\delta} \tag{33}$$

$$\le \frac{2}{\delta}\sqrt{\frac{1}{2}D_{KL}(p||p_\theta)} \tag{34}$$

$$\le \frac{\sqrt{2\epsilon}}{\delta}, \tag{35}$$

where $TV(p, p_\theta)$ is the total variation distance between p and p_θ. Eq. 34 is obtained using Pinsker's inequality. This inequality means that with $1 - \frac{\sqrt{2\epsilon}}{\delta}$ probability, $|\Delta p(v)| := |p_\theta(v) - p(v)| \le \delta$.

Let δ be a small quantity and $p_- = \min_{i=1,\cdots,N}(p(v_i))$, since

$$\sum_j(1 + \frac{|\Delta p(v_j)|}{p_-}) \ge \sum_j(1 + \frac{\Delta p(v_j)}{p_j}) \ge \sum_j(1 - \frac{|\Delta p(v_j)|}{p_-}), \tag{36}$$

and

$$w_{\text{norm},i} = \frac{w_i}{\sum_{j=1}^N w_j} = \frac{(1 + \frac{\Delta p(v_i)}{p(v_i)})\tilde{w}_i}{\sum_{j=1}^N(1 + \frac{\Delta p(v_j)}{p(v_j)})\tilde{w}_j}, \tag{37}$$

thus

$$\frac{1 + \frac{\Delta p(v_i)}{p(v_i)}}{1 + \frac{|\Delta p(v_i)|}{p_-}}\tilde{w}_{\text{norm},i} \le w_{\text{norm},i} \le \frac{1 + \frac{\Delta p(v_i)}{p(v_i)}}{1 - \frac{|\Delta p(v_i)|}{p_-}}\tilde{w}_{\text{norm},i}. \tag{38}$$

By rearranging terms, we obtain

$$|w_{\text{norm},i} - \tilde{w}_{\text{norm},i}| \le \max\{|\frac{\Delta p(v_i)/p(v_i) - |\Delta p(v_i)|/p_-}{1 + |\Delta p(v_i)|/p_-}|, |\frac{\Delta p(v_i)/p(v_i) + |\Delta p(v_i)|/p_-}{1 - |\Delta p(v_i)|/p_-}|\} \tag{39}$$

$$\le |\Delta p(v_i)|\frac{|1/p(v_i)| + |1/p_-|}{1 - \delta/p_-} \tag{40}$$

$$\le C|\Delta p(v_i)| \tag{41}$$

$$\le C|\delta|. \tag{42}$$

Because we can merge the v_i values that are the same, we can assume $v_1 < v_2 < \cdots < v_N$. Suppose Quantile$(\alpha; \sum_{i=1}^N w_{\text{norm},i}\delta_{v_i}) = v_k$, which means

$$\sum_{i=i}^k w_{\text{norm},i} \ge 1 - \alpha, \sum_{i=i}^{k-1} w_{\text{norm},i} < 1 - \alpha. \tag{43}$$

With Eq. 42, we can get that

$$\sum_{i=i}^k \tilde{w}_{\text{norm},i} \ge 1 - \alpha - Ck\delta, \sum_{i=i}^{k-1} \tilde{w}_{\text{norm},i} < 1 - \alpha + Ck\delta. \tag{44}$$

We can let δ be small enough to make k satisfy

$$\sum_{i=i}^k \tilde{w}_{\text{norm},i} \ge 1 - \alpha, \sum_{i=i}^{k-1} \tilde{w}_{\text{norm},i} < 1 - \alpha, \tag{45}$$

which means Quantile$(\alpha; \sum_{i=1}^N w_{\text{norm},i}\delta_{v_i}) =$ Quantile$(\alpha; \sum_{i=1}^N \tilde{w}_{\text{norm},i}\delta_{v_i})$. □

B. Additional Experiment Results

B.1. Inference Time

We have accelerated the efficiency of inference. On one hand, by introducing post-training, the data distribution is already close to the desired distribution before inference. On the other hand, we have significantly accelerated the diffusion model's sampling time using DDIM (Song et al., 2021). The Table 5 below presents a comparison of the inference time between our method and the baseline on the Burgers' equation on A800 with 8 CPUs. It shows that our method achieves a moderate speed and is much faster compared to TREBI, which is also based on the diffusion model.

Table 5: **Inference time on the Burgers' equation on A800 with 8 CPUs.**

Methods	BC	PID	SL-Lag	MPC-Lag	CDT	TREBI	SafeDiffCon
Inference Time (min)	0.1351	0.1091	0.8842	26.2905	0.0890	13.5525	2.3575

B.2. Ablation Study

In this subsection, we provide results of ablation studies on the 1D Burgers' equation and the tokamak fusion reactor. We also compare SafeDiffCon and SafeDiffCon without post-training, inference-time fine-tuning, and the uncertainty quantile Q. The results in Table 6 still show that the absence of any module affects the model, causing it to fail to meet the safety constraints. Therefore, each module is effective.

Table 6: **Results of the ablation study on the tokamak fusion reactor.** We compare SafeDiffCon with SafeDiffCon w/o post-training, inference-time fine-tuning and the uncertainty quantile Q.

Methods	$\mathcal{J} \downarrow$	$\mathcal{R}_{\text{sample}} \downarrow$	$\mathcal{R}_{\text{time}} \downarrow$
SafeDiffCon	0.0094	0%	0%
w/o post-training	0.0153	10%	0.52%
w/o fine-tuning	0.0210	50%	7.74%
w/o Q	0.0269	28%	0.65%

The only exception is the result of SafeDiffCon without post-training in Table 7. Due to the high precision requirements of this control scenario, large-scale post-training may easily disrupt the dynamics originally learned by the model during training. Therefore, directly applying fine-tuning allows the model to adapt to specific control tasks without significant loss of accuracy, enabling both safe and precise control.

Table 7: **Results of the ablation study on the Burgers' equation.** We compare SafeDiffCon with SafeDiffCon w/o post-training, inference-time fine-tuning and the uncertainty quantile Q.

Methods	$\mathcal{J} \downarrow$	$\mathcal{R}_{\text{sample}} \downarrow$	$\mathcal{R}_{\text{time}} \downarrow$	$\mathcal{R}_{\text{point}} \downarrow$
SafeDiffCon	0.0016	0%	0%	0%
w/o post-training	0.0016	0%	0%	0%
w/o fine-tuning	0.0007	40%	20%	3%
w/o Q	0.0006	30%	10%	1%

B.3. Analysis on Parameters

We conduct experimental analysis on coverage probability α, the split ratio of training/calibration data, and the weight of objective γ. The Table 8, 9 and 10 show that they do not impact much on performance, indicating that the model is robust.

Table 8: **Impact of the coverage probability α on the tokamak fusion reactor.**

Split Ratio	$\mathcal{J} \downarrow$	$\mathcal{R}_{\text{sample}} \downarrow$	$\mathcal{R}_{\text{time}} \downarrow$
0.8	0.01017	0%	0%
0.85	0.0091	0%	0%
0.9	0.0094	0%	0%
0.95	0.0095	0%	0%

Table 9: **Impact of the split ratio of training/calibration data on the tokamak fusion reactor.**

α	$\mathcal{J} \downarrow$	$\mathcal{R}_{\text{sample}} \downarrow$	$\mathcal{R}_{\text{time}} \downarrow$
0.005	0.0124	4%	0.03%
0.01	0.0093	0%	0%
0.02	0.0094	0%	0%

C. Visualization of Experiment Results

C.1. 1D Burgers' Equation

In this section, we provide additional visualizations of the control results for the 1D Burgers' equation, as shown in Figure 4. In these figures, the top row represents the original trajectories corresponding to the control targets, while the bottom row displays the trajectories controlled by SafeDiffCon. It can be observed that SafeDiffCon successfully controls the trajectories, preventing boundary violations and guiding them to the desired final state.

Table 10: **Impact of the split ratio of the weight of objective γ on the 2D incompressible fluid.**

γ	$\mathcal{J} \downarrow$	$SVM \downarrow$	$\mathcal{R} \downarrow$
0.01	0.3548	0	0%
0.1	0.4953	0.004	2%
0.3	0.498	0.003	2%

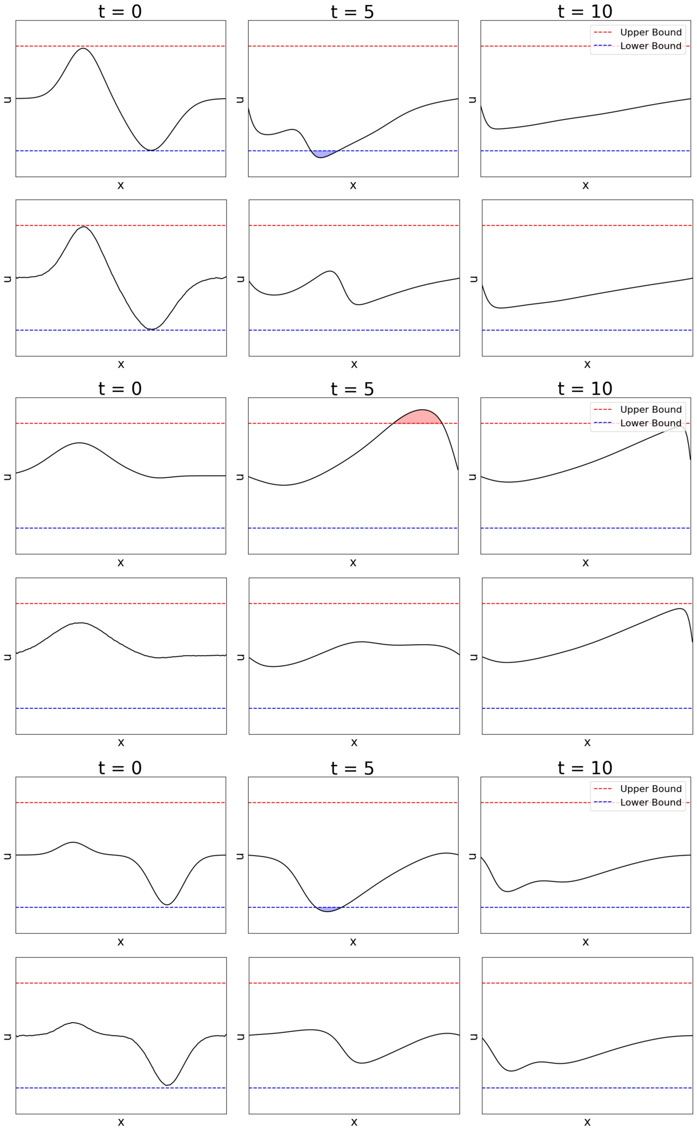

Figure 4: **Visualization of the 1D Burgers' equation.**

C.2. 2D Incompressible Fluid

Here, we provide additional visualizations of the control problems of 2D incompressible fluid. From the figures, we can observe that SafeDiffCon can successfully control the smoke to avoid the red hazardous region and reach the target bucket as well.

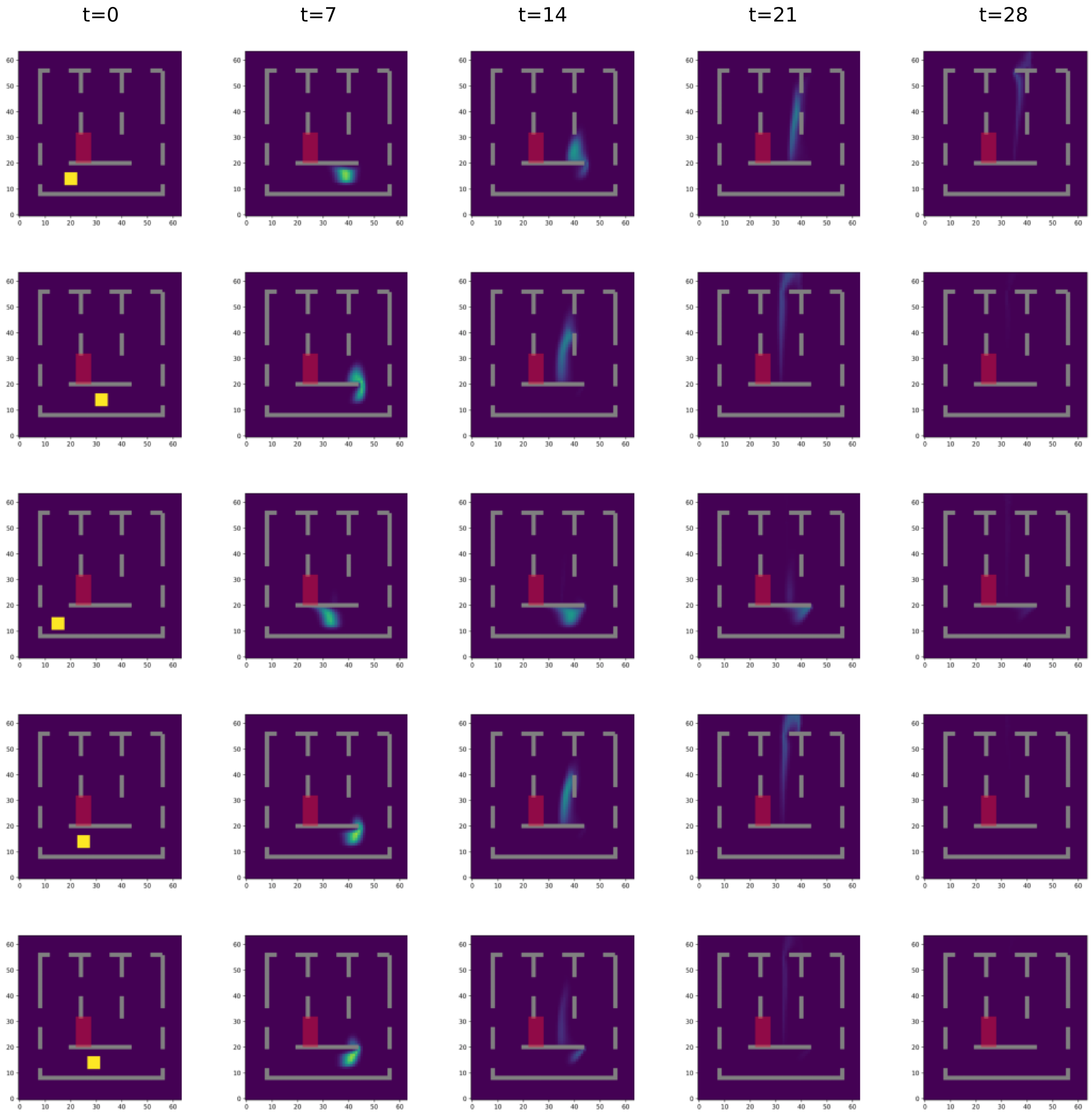

Figure 5: **Visualizations of the 2D incompressible fluid control problem.**.

D. Additional Details for Conformal Prediction

Conformal prediction is a flexible framework that provides prediction intervals with guaranteed coverage probabilities for new, unseen data points, under the assumption that the data are exchangeable.

Theoretical Foundations The exchangeability assumption is a cornerstone of conformal prediction. It requires that the order of the data points does not affect their joint distribution, meaning that any permutation of the indices yields an identical distribution. In particular, exchangeability holds for independent and identically distributed (i.i.d.) samples, a common assumption in machine learning tasks. Ensuring exchangeability guarantees the validity of the prediction intervals constructed using conformal prediction.

Implementation Details To implement conformal prediction, the dataset is first split into two subsets: a proper training set (D_{train}) and a calibration set (D_{cal}). A predictive model μ_θ is trained on the training set using a specified learning algorithm

$\mathcal{A}$. Once trained, the model generates predictions for the calibration set. These predictions are used to compute 'conformity scores', which measure the model's accuracy for each calibration point. Specifically, for each instance i in the calibration set, the conformity score S_i is defined as:

$$S_i = |\mu_\theta(X_i) - Y_i|, \quad i \in D_{\text{cal}}.$$

Additionally, a worst-case score of ∞ is included to account for extreme scenarios.

Then $1-\alpha$ quantile $q_{1-\alpha}(S)$ of the set of conformity scores is calculated, where α represents the desired significance level (*e.g.*, $\alpha = 0.05$ for 95% confidence interval).

Given a new data point X_{n+1}, the prediction interval for its corresponding output is calculated as:

$$\hat{C}_\alpha(X_{n+1}) = \left[\mu_\theta(X_{n+1}) - q_{1-\alpha}(S), \mu_\theta(X_{n+1}) + q_{1-\alpha}(S)\right].$$

This interval provides an estimate for the range within which the true value Y_{n+1} is expected to lie, with a coverage probability of at least $1-\alpha$. Thus, conformal prediction offers a flexible and robust method for constructing prediction intervals that account for both the model's accuracy and the variability in the data.

Theoretical Guarantees Conformal prediction provides theoretical guarantees for finite samples (Vovk et al., 2005; Lei et al., 2016). Specifically, for any new data point, the prediction interval satisfies the following probabilistic bound:

$$P(Y_{n+1} \in \hat{C}_\alpha(X_{n+1})) \geq 1 - \alpha.$$

This ensures that the true label Y_{n+1} will fall within the predicted interval at least $1-\alpha$ percent of the time. This framework, based on the assumption of exchangeability, provides a robust method for generating reliable prediction intervals, even in settings with limited sample sizes.

E. Details of the Method Implementation

E.1. Conditionally Sample $\mathbf{u}_\theta(\mathbf{w})$

In our proposed algorithm, we need to sample from the conditional distribution $p(\mathbf{u}|\mathbf{w})$ with the model that learns the joint distribution $p(\mathbf{u}, \mathbf{w})$. To achieve this, at each denoising step of the sampling process, we replace the noisy $\mathbf{w}$ in the input of the denoising network with the actual clean $\mathbf{w}$ that serves as the condition (Chung et al., 2023). In fact, this situation represents a special case of the data distribution that the denoising network encounters during training, where the $\mathbf{u}$ part is noisy, while the $\mathbf{w}$ part remains noise-free.

E.2. Algorithm of Post-training

Here we provide the specific algorithm of post-training in Algorithm 2.

E.3. Algorithm of Inference-time Fine-tuning

In this subsection, we provide the entire algorithm of inference-time fine-tuning in Algorithm 3.

Algorithm 2 Post-training of SafeDiffCon

```
Require Training set $D_{\text{train}}$, Calibration set $D_{\text{cal}}$, coverage probability $\alpha$, number of epochs $N$, number of updating steps per epoch $M$
for $n = 1, \dots, N$ do
    Compute the shifted score set $\tilde{\mathcal{S}}$ with $D_{\text{cal}}$ // Eq. 6
    Get the uncertainty quantile $Q(1-\alpha; \tilde{\mathcal{S}})$
    for $m = 1, \dots, M$ do
        Take gradient descent step on $\nabla_\theta \mathcal{L}_{\text{post-train}}$ with $D_{\text{train}}$ // Eq. 12
    end for
end for
return $\theta$
```

Algorithm 3 Inference of SafeDiffCon

```
Require Calibration set $D_{\text{cal}}$, coverage probability $\alpha$, number of iterations $N$
for $n = 1, \dots, N$ do
    Compute the shifted score set $\tilde{\mathcal{S}}$ with $D_{\text{cal}}$ // Eq. 6
    Get the uncertainty quantile $Q(1-\alpha; \tilde{\mathcal{S}})$
    Sample the control sequence $\mathbf{w}$ with guidance $\mathcal{G}$ // Eq. 14
    Take gradient descent step on $\nabla_\theta \mathcal{L}_{\text{fine-tune}}$ // Eq. 15
end for
Sample the control sequence $\mathbf{w}$ with guidance $\mathcal{G}$ // Eq. 14
return $\mathbf{w}$
```

F. Additional Details for 1D Burgers' Experiment

F.1. Experiment Setting

Following the previous works (Holl et al., 2020; Wei et al., 2024), we generate the 1D Burgers' equation dataset. This equation is formulated as follows:

$$\begin{cases} \frac{\partial \mathbf{u}(t,x)}{\partial t} = -\mathbf{u}(t,x) \cdot \frac{\partial \mathbf{u}(t,x)}{\partial x} + \nu \frac{\partial^2 \mathbf{u}(t,x)}{\partial x^2} + \mathbf{w}(t,x) & \text{in } [0,T] \times \Omega \\ \mathbf{u}(t,x) = 0 & \text{on } [0,T] \times \partial\Omega \\ \mathbf{u}(0,x) = \mathbf{u}_0(x) & \text{in } \{t=0\} \times \Omega, \end{cases} \tag{46}$$

where ν denotes the viscosity parameter, while $\mathbf{u}_0$ signifies the initial condition. We set $\nu = 0.01$, $T = 1$ and $\Omega = [0, 1]$.

During inference, alongside the control sequence $\mathbf{w}(t,x)$, our diffusion model generates states $\mathbf{u}(t,x)$. Our reported evaluation metric $\mathcal{J}$ is always computed by feeding the control $\mathbf{w}(t,x)$ into the ground truth numerical solver to get $\mathbf{u}_{\text{g.t.}}(t,x)$ and computed following Eq. (16). Following Eq. (5.1), we consider the safety constraint and define the safety score as s. In our experiment, the bound of safety score s_0 is set to 0.64, 89.7% of samples are unsafe among the training set, 90% of samples are unsafe among the calibration set and all of the samples in the test set are unsafe. The details of the 1D Burgers' equation dataset for safe PDE control problem are listed in Table 11.

Table 11: **Details of 1D Burgers' equation dataset.**

	Training Set	Calibration Set	Test Set
Unsafe Trajectories	34,985	900	50
Safe Trajectories	4,015	100	0

F.2. Model

The model architecture in this experiment follows the Denoising Diffusion Probabilistic Model (DDPM) (Ho et al., 2020). For control tasks, we condition on u_0, u_T and apply guidance to generate the full trajectories of $u_{[0,T]}$, $f_{[0,T]}$ and the safety score s. The hyperparameters for the 2D-Unet architecture are recorded in Table 12.

G. Additional Details for 2D Fluid Experiment

G.1. Experiment Setting

The 2D environment is modeled by the Navier-Stokes equation:

$$\begin{cases} \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} - \nu \nabla^2 \mathbf{v} + \nabla p = f, \\ \nabla \cdot \mathbf{v} = 0, \\ \mathbf{v}(0, \mathbf{x}) = \mathbf{v}_0(\mathbf{x}), \end{cases} \tag{47}$$

where $\mathbf{v}$ is the velocity, p is the pressure, f is the external force and ν is the viscosity coefficient.

Table 12: **Hyperparameters of 2D-Unet architecture in 1D experiment.**

Hyperparameter Name	Value
Initial dimension	128
Convolution kernel size	3
Dimension multiplier	[1,2,4,8]
Resnet block groups	1
Attention hidden dimension	32
Attention heads	4
Number of training steps	200000
DDIM sampling iterations	100
η of DDIM sampling	1

Following works from Holl et al. (2020); Wei et al. (2024); Hu et al. (2024), we use the package `PhiFlow` to generate the 2D incompressible fluid dataset. The control objective and data generation is the same as before (Wei et al., 2024). The main difference between our data and previous ones is that we consider the safety constraint here. We define the safety score as the percentage of smoke passing through a specific region. This reflects the need to limit the amount of pollutants passing through certain areas in real-world scenarios, such as in a watershed.

We simulate the fluid on a 128×128 grid. The selected hazardous region is $[44, 36] \times [40, 64]$. Since the optimal path for smoke, starting from a left-biased position, is likely to pass through this hazardous region, this poses a greater challenge for the algorithm: how to balance safety and achieving a more optimal objective, making this a more difficult problem.

G.2. Model

In this paper, the design of the three-dimensional U-net we use is based on the previous work (Ho et al., 2022). In our experiment, we utilize spatio-temporal 3D convolutions. The U-net consists of three key components: a downsampling encoder, a central module, and an upsampling decoder.

The diffusion model conditions on the initial density and uses guidance as previous mentioned to to generate the full trajectories of density, velocity, control, the objective $\mathcal{J}$ and the safety score. As $\mathcal{J}$ and s are scalers, we repeat them to match other channels. The hyperparameters for the 3D U-net architecture are listed in Table 13.

Table 13: **Hyperparameters of 3D-Unet architecture in 2D experiments**.

Hyperparameter Name	Value
Number of attention heads	4
Kernel size of conv3d	(3, 3, 3)
Padding of conv3d	(1,1,1)
Stride of conv3d	(1,1,1)
Kernel size of downsampling	(1, 4, 4)
Padding of downsampling	(1, 2, 2)
Stride of downsampling	(0, 1, 1)
Kernel size of upsampling	(1, 4, 4)
Padding of upsampling	(1, 2, 2)
Stride of upsampling	(0, 1, 1)
Number of training steps	200000
DDIM sampling iterations	100
η of DDIM Sampling	1
Intensity of guidance in control	100
Weight of safety term in guidance	10000

H. Additional Details for Tokamak Fusion Reactor

H.1. Experiment Setting

Following the previous work (Seo et al., 2022), the environment used in this experiment is a data-driven simulator designed to replicate the plasma behavior in the KSTAR tokamak. It is constructed using long short-term memory (LSTM) neural networks to capture the time-dependent dynamics of plasma, such as transport processes and flux diffusion. The LSTM-based model is trained on five years of experimental data from KSTAR, incorporating various tokamak control variables as inputs, including plasma current, toroidal magnetic field, line-averaged density, neutral beam injection and electron cyclotron heating powers, and boundary shape parameters.

The simulator predicts the evolution of key zero-dimensional (0D) plasma parameters ($1 - \alpha_p$, l_i, and q_{95}) on a 100 ms timescale in an autoregressive manner, using the current state and control inputs. This enables fast and experimentally relevant predictions of plasma responses. Using this model avoids the high costs and risks associated with real tokamak experiments while ensuring sufficient accuracy and efficiency for training.

I. Baselines

I.1. CDT

Constraints Decision Transformer (CDT) (Liu et al., 2023b) models control as a multi-task regression problem, extending the Decision Transformer (DT) (Chen et al., 2021). It sequentially predicts returns-to-go, costs-to-go, observations, and actions, making actions dependent on previous returns and costs. The authors propose two techniques to adapt the model for safety-constrained scenarios:

1. **Stochastic Policy with Entropy Regularization**: This technique aims to reduce the risk of constraint violations due to out-of-distribution actions. In a deterministic policy, the model selects a single action based on its learned policy, which may result in unsafe actions when faced with states not well represented in the training data. By using a stochastic policy, the model samples actions from a distribution, encouraging the exploration of a wider action space. Entropy regularization further enforces diversity in the sampled actions, making the model more robust in uncertain or underrepresented situations. This approach reduces the likelihood of selecting unsafe actions when faced with states outside the distribution of the training set.

2. **Pareto-Frontier-Based Data Augmentation**: The technique tries to resolve the conflict between maximizing returns and adhering to safety constraints by leveraging a Pareto-frontier of the training data. The Pareto-frontier consists of trajectories that provide the highest possible return under specific safety constraints. Fitting a polynomial to the Pareto-frontier helps identify conflicting high-return and safety constraint pairs, which are then used for augmentation. The augmentation generates synthetic trajectories by relabeling safe trajectories from the Pareto-frontier with higher returns and assigning higher or equal safety constraints. This encourages the model to imitate the most rewarding, safe trajectories when the desired return given the safety constraint is infeasible.

In the 2D incompressible fluid experiment, the model fails to extrapolate safe trajectories with higher returns due to the training data **covering a broad range of costs, while the desired cost lies in a narrow range**. The augmentation treats all safe trajectories on the Pareto-frontier equally, without emphasizing the region of interest. This complexity, however, could potentially be addressed by SafeDiffCon, which is indicated in Table 2. We use the official CDT implementation and follow DT guidelines to sweep desired returns and cost constraints in testing time. In the 1D Burgers experiment, we modify the control objective from the original mean squared error $\mathcal{J}$ between the prediction and target, to an exponential form $\exp(-\mathcal{J})$. This new objective is bounded within $[0, 1]$, which better aligns with the reward-maximizing setup in reinforcement learning used by CDT. For the 2D incompressible fluid setup, the state, action, and cost prediction heads each consist of 3-layer MLPs with the transformer's hidden dimension as the inner size.

I.2. BC-All

The Behavior Cloning (BC) algorithm, introduced by (Pomerleau, 1988), is a foundational technique in imitation learning. BC is designed to derive policies directly from expert demonstrations, utilizing supervised learning to associate states with corresponding actions. This method eliminates the necessity for exploratory steps commonly required in reinforcement

Table 14: **Hyperparameters of 1D CDT**.

1D Burgers'	All data	Safe filtered
State Dimension	256	256
Action Dimension	128	128
Hidden Dimension	1024	1024
Number of Transformer Blocks	2	2
Number of Attention Heads	8	8
Horizon (Sequence Length)	5	5
Learning Rate	1e-4	1e-4
Batch Size	64	64
Weight Decay	1e-5	1e-5
Learning Steps	1,000,000	1,000,000
Learning Rate Warmup Steps	500	500
Pareto-Frontier Fitted Polynomial Degree	0	4
Augmentation Data Percentage	0.3	0.3
Max Augment Reward	10.0	10.0
Min Augment Reward	1.0	1.0
Target Entropy	-128	-128
Testing Time Sweep Returns	9.0, 9.9	9.0, 9.9
Testing Time Sweep Costs	0.0, 1.0, 2.0, 3.0	0.0, 1.0, 2.0, 3.0

learning by replicating the actions observed in expert demonstrations. One of the significant advantages of BC is that it does not involve interacting with the environment during the training phase, which streamlines the learning process and diminishes the demand for computational resources.

In this approach, a policy network is trained using standard supervised learning strategies aimed at reducing the discrepancy between the actions predicted by the model and those performed by the expert in the dataset. The commonly used loss function for this purpose is the mean squared error between the predicted actions and expert actions. The dataset for training comprises state-action pairs harvested from these expert demonstrations. In the work, we employ the implementation as (Liu et al., 2023a).

I.3. BC-Safe

Following the baseline in (Liu et al., 2023a), the BC-Safe is fed with only safe trajectories filtered from the training dataset, satisfies most safety requirements, although with conservative performance and lower rewards. Others are same as BC-All, except for the safe trajectories.

I.4. SL-Lag

Hwang et al. (2022) introduces a supervised learning (SL) based method to control PDE systems. It first trains a neural surrogate model to capture the PDE dynamics, which includes a VAE to compress PDE states and controls into the latent space and another model to learn the PDE's time evolution in the latent space. To obtain the optimal control sequence, SL can compute the gradient $\nabla_{\mathbf{w}}\mathcal{J}$, where $\mathcal{J}$ is the control objective and f is the input control sequence. Then iterative gradient optimization can be executed to improve the control sequence.

To ensure that the optimal control is compatible with the hard constraint in our experiments, we follow Chow et al. (2018) to apply the Lagrange optimization method to the constrained optimization. Specifically, we iteratively solve the optimization problem below:

$$\max_{\lambda\geq 0}\min_{\mathbf{w}} \mathcal{J}(\mathbf{w}) + \lambda(s(\mathbf{u}(\mathbf{w})) - c). \tag{48}$$

We denote the modified SL method SL-Lag.

Table 15: **Hyperparameters of 2D CDT**.

2D incompressible fluid	All data	Safe filtered
State Dimension	$3\times64\times64$	$3\times64\times64$
Action Dimension	$2\times64\times64$	$2\times64\times64$
Hidden Dimension	512	512
Number of Transformer Blocks	3	3
Number of Attention Heads	8	8
Horizon (Sequence Length)	10	10
Learning Rate	1e-4	1e-4
Batch Size	8	8
Weight Decay	1e-5	1e-5
Learning Steps	1,000,000	1,000,000
Learning Rate Warmup Steps	500	500
Pareto-Frontier Fitted Polynomial Degree	4	4
Augmentation Data Percentage	0.3	0.3
Max Augment Reward	32.0	32.0
Min Augment Reward	1.0	1.0
Target Entropy	$-(2\times64\times64)$	$-(2\times64\times64)$
Testing Time Sweep Returns	18.0, 32.0	18.0, 32.0
Testing Time Sweep Costs	0.0, 0.1, 0.2	0.0, 0.1, 0.2

Table 16: **Hyperparameters of network architecture and training for SL-Lag in 1D Burgers' experiment**.

Hyperparameter name	Value
Initialization value of $\mathbf{w}$	0.001
Optimizer of $\mathbf{w}$	LBFGS
Learning rate of $\mathbf{w}$	0.1
Initialization value of the Lagrange multiplier λ	0
Optimizer of λ	SGD
Learning rate of λ	10
Iteration of λ	2
Loss function	MSE

I.5. MPC-Lag

We adopt the MPC with stochastic gradient descent and choose the planning step $K = 1$. The simulation model is the same as the one in SL-Lag. Also, the way we combine MPC with the Lagrangian method is also the same as SL-Lag. The hyperparameters are reported in Table 17.

I.6. TREBI

In Lin et al. (2023), the diffusion model is adopted for the planning task under safety budgets. It generates trajectory under safety constraints using classifier-guidance Dhariwal & Nichol (2021) by adding a safety loss to the reward guidance following Diffuser Janner et al. (2022).

However, the original setting is different from our experiments. In our 1D Burgers' equation control, our objective is that a certain state equals the target state, which in-painting diffusion condition Janner et al. (2022) is more appropriate for. Furthermore, as in our method and in (Wei et al., 2024), a conditional diffusion model can be learned to tackle the objective more directly. In addition, TREBI follows the setting in Diffuser where the interaction with the environment is allowed which in our experiments becomes an MPC method. Note that the reported results of our method do not involve interaction with the surrogate model (though our method can easily adapted to be an MPC method). Thus, the results of TREBI in Table 1 and 2 have an unfair advantage.

Table 17: **Hyperparameters of for MPC-Lag in 1D Burgers' experiment**.

Hyperparameter name	Value
Initialization value of $\mathbf{w}$	0.01
Optimizer of $\mathbf{w}$	LBFGS
Learning rate of $\mathbf{w}$	0.01
Initialization value of the Lagrange multiplier λ	0
Optimizer of λ	SGD
Learning rate of λ	10
Iteration of λ	2
Loss function	MSE

Therefore, for 1D Burgers' experiment, we conducted different experiments on TREBI including (1) planning multiple times with interaction with the surrogate model or (2) planning only once, and with target state conditioning or target state guidance. The target state guidance + planning multiple times turned out the best and is reported in Table 1. For 2D smoke control, the target is not a state constraint but a reward, and the planning multiple-step setting is too computationally expensive. To this end, we use reward guidance with planning one single time, which is identical to the ablation study of our method in Table 4. The hyperparameters of the 1D experiment are reported in Table 18, and those of 2D are in Table 19.

Table 18: **Hyperparameters of network architecture and training for TREBI in 1D Burgers' experiment**.

Hyperparameter name	Value
Reward guidance intensity	50
Safety guidance intensity	1
Cost budget	0.64
Number of guidance steps	10
Denoising steps	200
Sampling algorithm	DDPM
U-Net dimension	64
U-Net dimension mltiplications	1, 2, 4, 8
Planning horizon	8 steps
Optimizer	Adam
Learning rate	0.0002
Batch size	16
Loss function	MSE

I.7. PID

Proportional Integral Derivative (PID) control (Li et al., 2006) is a versatile and effective method widely employed in numerous control scenarios. For 1D control task, we mainly implement the PID baseline adapted from (Wei et al., 2024). More detailed configurations can be found in Table 20

Table 19: **Hyperparameters of network architecture and training for TREBI in 2D Incompressible Fluid Control experiment**.

Hyperparameter name	Value
Reward guidance intensity	5000
Safety guidance intensity	0.01
Cost budget	0.1
Number of guidance steps	1
Denoising steps	20
Sampling algorithm	DDPM
3D U-Net dimension	8
3D U-Net dimension mltiplications	1, 2, 4, 8
Planning horizon	8 steps
Optimizer	Adam
Learning rate	0.0002
Batch size	16
Loss function	MSE

Table 20: **Hyperparameters of network architecture and training for ANN PID**.

Hyperparameter name	Value
Kernel size of conv1d	3
Padding of conv1d	1
Stride of conv1d	1
Activation function	Softsign
Batch size	16
Optimizer	Adam
Learning rate	0.0001
Loss function	MAE