HOFLON: Hybrid Offline Learning and Online Optimization for Process Start-Up and Grade-Transition Control

Alex Durkin[1], Jasper Stolte[2], and Mehmet Mercangöz[1]

[1]Department of Chemical Engineering, Imperial College London, SW7 2AZ, UK
[2]Shell Information Technology International BV, 1031 HW Amsterdam, NL

Abstract

Start-ups and product grade-changes are critical steps in continuous-process plant operation, because any misstep immediately affects product quality and drives operational losses. These transitions have long relied on supervision by a handful of expert operators, but the progressive retirement of that workforce is leaving plant owners without the tacit know-how needed to execute them consistently. In the absence of a process model, offline reinforcement learning (RL) promises to capture—and even surpass—human expertise by mining historical start-up and grade-change logs, yet standard offline RL struggles with distribution-shift and value-overestimation whenever a learned policy ventures outside the data envelope. We introduce HOFLON (Hybrid Offline Learning + Online Optimization) to overcome those limitations. Offline, HOFLON learns (i) a latent data manifold that represents the feasible region spanned by past transitions and (ii) a long-horizon Q-critic that predicts the cumulative reward from state–action pairs. Online, it solves a one-step optimization problem that maximizes the Q-critic while penalizing deviations from the learned manifold and excessive rates of change in the manipulated variables. We test HOFLON on two industrial case studies—a polymerization reactor start-up and a paper-machine grade-change problem—and benchmark it against Implicit Q-Learning (IQL), a leading offline-RL algorithm. In both plants HOFLON not only surpasses IQL but also delivers on average better cumulative rewards compared to the best start-up or grade-change ever observed in the historical data, demonstrating its potential to automate transition operations beyond current expert capability.

1 Introduction

Start-up and product grade-change operations are critical transient phases in continuous process plants. These differ fundamentally from maintaining a stable steady-state by exhibiting a definable initial condition and a conclusive final state, thereby requiring distinct dynamic control strategies (Krüger et al., 2004). Any misstep during these transients immediately affects product quality, causes off-spec material, and leads to economic losses. For example, polymer reactors experience considerable off-spec polymer during transitions due to nonlinear reactor behavior (Kao et al., 2025). Achieving rapid, safe, and constraint-compliant transitions provides a significant competitive advantage. Historically, these operations relied on tacit knowledge of experienced operators, which is difficult to document and standardize. However, many senior operators are retiring, creating a significant knowledge gap potentially leading to operational variability and safety risks. Consequently, there is a pressing need to codify existing knowledge systematically to help junior process operators or to enable autonomous operations.

Procedural automation and state-based control have been widely adopted to ensure consistent execution of start-up operations in steps but these procedures cannot capture complex control trajectories (Chang et al., 2021). Advanced control approaches such as Model Predictive Control (MPC) and dynamic optimization have also been employed to optimize grade transitions in polymer and paper industries. However, these approaches require accurate mechanistic models, which are challenging and resource-intensive to develop and might not be always feasible to deploy (Chu, 2019).

Grade transitions can be viewed at two interacting decision layers. The scheduling/planning layer determines when transitions occur and which grade-to-grade changes are executed, trading demand and inventory objectives against changeover losses (Mostafaei et al., 2020). At the control layer, given a specified transition, the objective is to execute the change quickly and safely while minimizing off-spec production and excessive actuation. This work focuses on the control layer, assuming the transition schedule is given. From a scheduling perspective, losses can often be reduced by sequencing similar grades consecutively; conversely, unavoidable large grade jumps place greater demands on the transition controller (Choi et al., 2021).

1.1 Industrial examples

Polymerization reactors Polymerization reactor start-ups and grade switches are inherently nonlinear, multivariable operations; if executed slowly or inaccurately they generate large volumes of off-spec product and erode profit (Bonvin et al., 2005; Lee and Lee, 2003). Because each minute of transition time produces waste, operators sometimes overshoot or undershoot set-points to accelerate the change, risking constraint violations and safety incidents (Prata et al., 2008). Hence grade transitions impose a dual burden: complex dynamics with tight constraints, and the economic drive to minimise downtime and scrap. Steady-state quality alone is insufficient—manufacturers must also perform grade changes quickly and reliably to meet market demands (Bonvin et al., 2005). Early solutions relied on first-principles modeling and optimal-control formulations that computed feed, hydrogen, and temperature trajectories to minimize time or off-spec volume (Prata et al., 2008). Model predictive control (MPC) soon became the implementation vehicle, coordinating multivariable actions within safety limits. Full-scale MPC/APC deployments in polyethylene plants, for example, report 25-50% faster transitions, lower scrap, and roughly 7% higher throughput (Bonvin et al., 2005). More recently, data-centric methods augment or replace physics-based approaches. Iterative learning control (ILC) refines repeated grade changes, shrinking error and waste run by run (Bristow et al., 2006; Lee and Lee, 2003). Machine-learning models—especially hybrid neural-network "digital twins" that blend mechanistic balances with data-driven kinetics—feed into nonlinear MPC and provide virtual sensors for on-the-fly quality prediction (Daoutidis et al., 2024; Chang et al., 2022; BenAmor et al., 2004). These adaptive strategies push grade-change automation toward shorter, safer, and more economical transitions while reducing reliance on perfect first-principles knowledge. In the absence of any models purely data-driven approaches can offer a solution as polymerization plants operate in multi-day to multi-week campaigns per grade generating sufficient data (Larsson et al., 2012).

Paper machines Early work on grade changes focused on dynamic modeling of the process during transitions (S. C. Chen, 1995), enabling the first optimization-based actuator and set-point trajectories (H. Ihalainen and R. Ritala, 1996). With the rise of multivariable predictive control, industrial case studies showed that coordinated, model-based transition control can shorten change times by up to 30% and cut variability in basis weight and moisture (Timothy F. Murphy and Shih-Chin Chen, 1999). Recent advances go further: data-driven schedulers minimize *how often* grade changes occur by optimizing production sequences (Hossein Mostafaei et al., 2020), while nonlinear MPC packages inside modern quality-control systems provide operator-friendly one-button transitions on the machine floor. Field reports from large mills confirm that combining well-maintained grade-change libraries with continuous monitoring sustains 28-35% faster transitions over the long term (Chu, 2019). Despite these automation gains, experienced operators remain pivotal: they validate soft-sensor predictions, intervene during upsets, fine-tune set-points between grades, and supply the knowledge that shapes and updates transition libraries(Vipetec Ltd., 2025). Depending on the product portfolio, paper mills could change grades daily with specialty operations reporting multiple changes per day leading to rich data, which can be used for purely data-driven control approaches (Papagrigoraki and Leberle, 2015).

1.2 Offline reinforcement learning

Reinforcement learning is the field within machine learning in which an agent learns to optimize its behavior in an environment through interaction (Sutton and Barto, 2018). The key advantage of reinforcement learning is that it can learn a control policy which deals with the trade off between greediness and patience to chase an even greater expected reward in the future. The theoretical framework underpinning reinforcement learning

is the Markov Decision Process (MDP). From a process systems perspective, one major challenge for RL in process plants is the development of realistic and representative simulation environments that capture the nonlinear, multivariable nature of chemical processes, including reaction kinetics, heat and mass transfer, and operational constraints. Safety and stability requirements further complicate the learning process, as policies must not only optimize performance but also ensure that operating conditions remain within safe and feasible bounds (Baldea et al., 2025).

In offline reinforcement learning (Levine et al., 2020; Prudencio et al., 2023) a buffer of historic observations is used to train a policy for the agent. In this setting the data was collected under an unknown behavior policy π_β, which could be any combination of manual operation with supervisory control. The objective is to learn a safe and robust control policy from this data without further environment interaction.

The key challenge with offline reinforcement learning is *distributional shift* (Kumar et al., 2019), which can be conceptually understood as the difficulty to accurately estimate the value of state/action combinations that were not observed in the historic data. To illustrate, consider the temporal difference loss to minimize the Bellman error for the Q-function as given below:

$$\mathcal{L}(\theta) = \mathbb{E}_{(s,a,s')\sim\mathcal{D}}\left[\left(r(s,a) + \gamma \max_{a'} Q_{\hat{\theta}}(s',a') - Q_\theta(s,a)\right)^2\right] \tag{1}$$

where $\mathcal{D}$ is the dataset of (state, action, next state) observations, Q_θ is the parameterized Q-function prediction for the current state-action pair, and $r(s,a) + \gamma \max_{a'} Q_{\hat{\theta}}$ is the Bellman equation for calculating the target Q-value as the sum of the immediate reward and the discounted maximum expected future rewards.

The challenge with distributional shift is that the loss in equation 1 contains a term $Q_{\hat{\theta}}(s',a')$ which requires evaluating the Q-function at state/action combinations that are not present in the data. Even worse, this Q-function evaluation is maximized over all possible future actions leading to gross value overestimation, learning policies that exploit this non-existing value.

Typical strategies to stabilize learning as employed in online reinforcement learning include double Q-learning (van Hasselt et al., 2015) and using target networks (Mnih et al., 2015) or simply using vast amounts of training data. These strategies can only slightly help reduce the magnitude of the value overestimation but do not fundamentally solve the problem. Since this issue was clearly articulated in (Kumar et al., 2019) in 2019 the field of offline reinforcement learning gathered momentum and many strategies have been devised to develop policies with more robustness towards distributional shift. Algorithms such as Conservative Q-Learning (CQL) (Kumar et al., 2020), Implicit Q-Learning (IQL) (Kostrikov et al., 2021), and Behavior-Regularized Actor-Critic (BRAC) (Wu et al., 2019) mitigate distributional shift by constraining policy updates toward the behavior policy or learning conservative value estimates.

Implicit Q-learning (IQL) (Kostrikov et al., 2021) is an offline RL approach that avoids the distributional shift problem by never querying the target Q-function for state/action combinations that are not in the dataset. This fully in-sample learning depends on an approximation of the Q-value of the next state/action combination by a state-value function. Expectile regression is used to predict an upper expectile of the state-value function with respect to the action distribution:

$$\mathcal{L}_V(\psi) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[\mathcal{L}_2^\tau\left(Q_{\hat{\theta}}(s,a) - V_\psi(s)\right)\right] \tag{2}$$

where $\mathcal{L}_2^\tau(u) = |\tau - \mathbb{1}(u<0)|u^2$ is the expectile loss function, which is a generalization of the mean squared error loss which enables asymmetric weightings to favour larger expectiles. If τ is equal to 0.5, the expectile regression is identical to the normal MSE. As τ is increased towards 1 the value no longer learns the average value of a given state over the action distribution but approaches the maximum value supported by the actions in the data. This estimated value function is then used to backup into the Q-function update instead of $Q_{\hat{\theta}}(s',a')$:

$$\mathcal{L}_Q(\theta) = \mathbb{E}_{(s,a,s')\sim\mathcal{D}}\left[\left(r(s,a) + \gamma V_\psi(s') - Q_\theta(s,a)\right)^2\right] \tag{3}$$

Once an estimated Q-value function is learned, the policy maximizing the expected value is extracted using advantage weighted regression (AWR) (Peng et al., 2019), which also only evaluates the Q-function at state/action combinations that are present in the dataset:

$$\mathcal{L}_\pi(\phi) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[\exp(\beta(Q_{\hat{\theta}}(s,a) - V_\psi(s)))\log\pi_\phi(a|s)\right] \tag{4}$$

where β is an inverse temperature parameter which controls the trade-off between maximizing the Q-function ($\beta \to \infty$) and staying close to the behavior policy ($\beta \to 0$) thereby supporting a spectrum of risk sensitivities. AWR is based on a forward KL divergence regularization pulling the policy towards actions observed in the data, pulling more strongly towards actions of higher estimated value.

Early generative modeling approaches, particularly autoencoders, have been widely utilized to constrain policy outputs within the training dataset's action distribution. For instance, Batch-Constrained Q-learning (BCQ) (Fujimoto et al., 2019) employs a conditional variational autoencoder (VAE) trained on the offline dataset, restricting action selection to within dataset support and preventing unrealistic extrapolations. Similarly, Policy in Latent Action Space (PLAS) (Zhou et al., 2020) leverages a latent action representation learned via a VAE, ensuring that generated actions remain inherently bounded within the support of available data. While these autoencoder-based methods effectively mitigate OOD queries, their conservative nature can also limit beneficial policy improvements due to overly strict constraints.

Recently, diffusion models have emerged as powerful alternatives capable of capturing complex, multi-modal action distributions. Diffusion-based methods, such as Diffusion-QL (Wang et al., 2023), learn action distributions through iterative denoising processes, inherently generating actions that closely adhere to the dataset distribution. However, these methods initially exhibited high computational overhead. Subsequent advancements like Efficient Diffusion Policies (EDP) (Kang et al., 2023) significantly reduced training complexity by reconstructing actions from partially denoised states. Trajectory-level approaches, exemplified by Diffuser (Janner et al., 2022), extend this concept by planning entire action sequences conditioned on desired outcomes, thus inherently curbing the occurrence of OOD actions.

Current research directions further enhance diffusion models by integrating value-aware guidance mechanisms. Advantage-Weighted Diffusion Actor-Critic (ADAC) (Chen et al., 2025) combines diffusion-based action generation with advantage-weighted value updates, selectively promoting potentially advantageous OOD actions while penalizing harmful ones. Diffusion-DICE (Mao et al., 2024) introduces a two-stage guidance process, first generating candidate actions within the dataset distribution and subsequently selecting optimal candidates based on critic evaluations, thereby effectively minimizing OOD errors. Additionally, prior-guided diffusion planning methods (Ki et al., 2025) replace standard Gaussian priors with learned priors concentrating diffusion processes on high-value, in-distribution trajectories. Collectively, these methods illustrate a principled evolution toward robust offline RL by leveraging generative modeling frameworks to balance action conservatism with the capacity for meaningful policy improvement.

1.3 Q-Learning and direct optimization of the action–value function for continuous control

A central challenge in applying reinforcement learning to problems with continuous action spaces is the selection of the optimal action. Given a learned action-value (or Q-) function, $Q(s, a)$, which estimates the long-term return from taking action a in state s, the agent must find the action that maximizes this function at each decision step. The dominant paradigm for addressing this challenge is the **actor-critic framework** (Lillicrap et al., 2016). In this approach, a separate "actor" network, $\pi_\phi(s)$, is trained to approximate the $\arg\max_a Q_\theta(s, a)$ operation. At runtime, action selection is a fast, single forward pass through the actor network. While computationally efficient, this introduces a second, interdependent learning problem that is notoriously difficult to stabilize. More critically for offline learning, the actor can become detached from the critic, learning to exploit regions where the Q-function erroneously predicts high values due to extrapolation error beyond the support of the offline dataset. An alternative and increasingly powerful paradigm is to forgo the actor network entirely and instead perform **direct optimization of the Q-function** at runtime. In this *critic-only* approach, the learned critic $Q_\theta(s, a)$ is used directly as a decision-time objective. At each time step, the current state s_k is observed and treated as fixed, and the control move is obtained by solving a one-step optimisation problem over the admissible action set:

$$a_k^\star = \arg\max_{a \in \mathcal{A}} Q_\theta(s_k, a). \tag{5}$$

This perspective is particularly attractive in offline and safety-critical settings, as it avoids committing to a single parametric policy and enables the incorporation of additional constraints or regularizes at deployment time. First, it eliminates the instabilities associated with training a separate actor. Second, it provides a

natural mechanism for enforcing constraints on the action space. Third, and most relevant to our work, it allows for the inclusion of additional objectives and regularizers directly into the action-selection process, enabling fine-grained control over the policy's behavior. The primary challenge of this approach is the non-convex nature of the maximization problem itself. A rich body of recent work has focused on developing effective methods to solve or approximate this optimization.

Tackling the Maximization Challenge. Research into direct Q-function optimization has explored several distinct strategies. One direction seeks to make the optimization tractable by design. **Structured parameterizations** impose constraints on the Q-function's architecture; for example, Normalized Advantage Functions (NAF) use a quadratic form for the advantage to yield an analytical solution (Gu et al., 2016), while Input-Convex Neural Networks ensure that any local optimum found is global (Amos et al., 2017). When the Q-function is an unconstrained deep neural network, the maximization can be addressed with **stochastic search and sampling methods**. QT-Opt successfully used a cross-entropy method (CEM) for robotic grasping without any actor network (Kalashnikov et al., 2018). Soft Q-Learning (SQL) reinterprets the Q-function as an energy landscape and samples high-value actions from the corresponding Boltzmann distribution, avoiding the hard maximization entirely (Haarnoja et al., 2017). For applications where optimality is critical, **exact mathematical programming** offers a powerful solution. If the Q-function is a ReLU network, the maximization can be formulated as a mixed-integer linear program (MILP) and solved to global optimality, producing highly accurate targets and superior control performance (Ryu et al., 2020; Yuan et al., 2023). These articles demonstrates that by framing action selection as an explicit optimization problem, we gain the ability to incorporate domain knowledge and safety considerations for the solution. This paradigm provides an important element for the approach we present in this paper.

The approach we present in this paper is designed for a common yet challenging scenario in industrial control: we lack a first-principles dynamic model or a high-fidelity simulator, but have access to a rich historical dataset of past operations. Critical procedures like process start-ups and product grade-changes often fall into this category, where logs capture the operational history, whether generated by experienced human operators or existing control systems. The core opportunity lies in leveraging this data not to imitate past behavior, but to synthesize a novel control strategy that provides a better performance than historical runs by intelligently recombining learned sub-policies.

1.4 Contributions

This paper makes the following primary contributions to data-driven process control and offline reinforcement learning:

- We introduce a hybrid framework HOFLON, which learns a long-horizon critic (value model) offline and computes actions at runtime by solving a regularized optimization problem over the admissible action space. This design exposes the performance–conservatism–smoothness trade-offs explicitly at deployment time and avoids relying on an actor network to generalize under distribution shift.
- To mitigate out-of-distribution action selection, we learn a compact model of data support in the joint state–action space using an adversarially trained autoencoder. The resulting latent reconstruction score is incorporated online as an explicit regularization term, discouraging actions that are poorly supported by the offline dataset while still allowing optimization-based improvement within the demonstrated operating envelope.
- We validate HOFLON on two challenging transition problems: a nonlinear polymerization reactor start-up and a multivariable paper-machine grade change. Across both benchmarks, HOFLON consistently improves transition performance relative to a representative offline RL baseline (IQL) and relative to the best trajectories contained in the available offline logs, while maintaining smooth actuation through an explicit move-suppression term. Code and datasets are released to enable reproducible comparison and future benchmarking.

The remainder of this paper is organized as follows. Section 2 details the HOFLON algorithm. Section 3 describes the process benchmarks. Section 4 presents our experimental results, followed by a discussion in Section 5 . Finally, Section 6 concludes the paper.

2 Methodology

A controller for start-ups and grade transitions must explicitly balance three competing requirements: (i) maximizing long-horizon performance, (ii) keeping actions within a safe operating envelope supported by available data, and (iii) enforcing smooth actuator movements to limit wear and avoid aggressive transients. Most standard offline RL methods pursue these requirements only indirectly, by regularizing a single monolithic policy during training. In practice, this makes it difficult to tune the performance–safety–smoothness trade-offs in a transparent way.

HOFLON is built around a principle of explicit decomposition. Rather than learning a single policy, we learn two complementary, interpretable components offline: a Q-critic that captures the long-term value of state–action pairs, and a data-manifold model that quantifies whether a candidate action is supported by the offline dataset for the current state. Both are learned as functions of state and action, preserving their joint dependence. During online deployment, the current state is observed and treated as a fixed parameter, which turns these learned state–action maps into action-space objectives evaluated at the present state. We then compute the control move by solving a real-time optimization problem that combines (i) critic value maximization, (ii) a penalty that discourages out-of-distribution actions via the manifold model, and (iii) an explicit regularizer on the rate of change of actions to promote smooth operation. This structure separates value, data-supported safety, and smoothness into independently tunable terms, enabling robust decision-making and clearer operational interpretation. Figure 1 provides a graphical overview of the HOFLON framework.

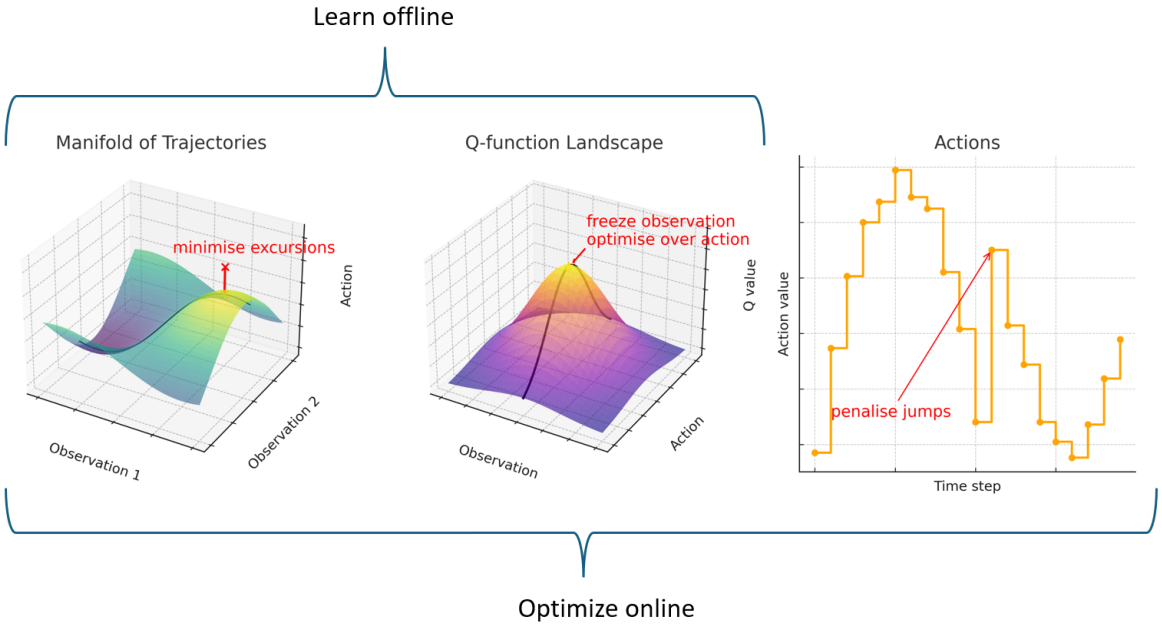

Fig. 1: Conceptual representation of the elements in the HOFLON-RL algorithm.

We introduce a novel control algorithm, **H**ybrid **O**ffline **L**earning with **ON**line **R**einforcement **L**earning (**HOFLON-RL**). This method synthesizes a data-driven control policy by combining offline training on historical data with real-time, solver-based optimization. The architecture is designed for complex, continuous-action control problems where safety and performance are paramount, such as industrial process control. It avoids the instabilities of traditional actor-critic methods by directly optimizing a learned action-value function, regularized by a model of the safe operating envelope.

2.1 Problem formulation

We model the controller-plant interaction as a Markov Decision Process (MDP), defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, r, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the continuous action space, $p(s_{k+1}|s_k, a_k)$ is the unknown state transition dynamics, $r(s_k, a_k)$ is the reward function, and $\gamma \in (0, 1)$ is the discount factor.

At each discrete time step k, the controller observes the plant state $s_k \in \mathcal{S} \subset \mathbb{R}^d$ and selects an action $a_k \in \mathcal{A} \subset [0, 1]^m$. The actions are scaled to physical units before being applied to the plant. The system then transitions to a new state s_{k+1} and yields a reward r_k. The objective is to find a policy $\pi(a_k|s_k)$ that maximizes the expected discounted return $G_k = \mathbb{E}\left[\sum_{j=0}^{\infty} \gamma^j r_{k+j}\right]$, subject to two critical constraints:

1. **Actuation Limits:** Actions must remain within hard physical bounds, $a_k \in [0, 1]^m$.
2. **Distributional Constraint:** The policy must only select actions that keep the system within the state-action distribution of the available historical data, ensuring safe and predictable behavior.

2.2 HOFLON-RL framework

The HOFLON-RL framework is divided into two distinct phases: an offline learning phase where historical data is used to train two key models, and an online optimization phase where these models are deployed to make real-time control decisions.

2.2.1 Offline learning phase

Given a static, offline dataset of historical transitions $\mathcal{D} = \{(s_k, a_k, r_k, s_{k+1})\}_{k=1}^{N}$, collected under one or more unknown behavioral policies, we learn two components: an action-value function (the Q-critic) that predicts long-term returns, and a generative model (the AAE) that captures the data distribution.

1. Learning the action-value function (Q-Critic). To guide the online optimization towards high-return actions, we train a Q-critic network, $\hat{Q}_\theta(s, a)$, to approximate the true action-value function $Q^*(s, a)$. Many offline RL algorithms learn the Q-function via Bellman bootstrapping. However, for finite-horizon episodic tasks common in grade transitions and startups, we can directly estimate the value function by regressing against the total empirical return accessible in historical data, which leads to stable solutions.

First, we compute the discounted return for each trajectory in the dataset:

$$G_k = \sum_{j=0}^{H-1} \gamma^j r_{k+j}, \tag{6}$$

where H is the episode horizon. The Q-critic $\hat{Q}_\theta$ is then trained via supervised learning to minimize the mean squared error:

$$\mathcal{L}_Q(\theta) = \mathbb{E}_{(s_k, a_k, G_k)\sim\mathcal{D}}\left[\left(G_k - \hat{Q}_\theta(s_k, a_k)\right)^2\right]. \tag{7}$$

The input to the Q-critic is an augmented state-action vector $z_k = [s_k, e_k, I_k, a_k]$, where $e_k = h(s_k, s_{\text{sp}})$ is the tracking error and $I_k = \sum_{i=0}^{k} e_i \Delta t$ is the integrated error, providing the critic with task-relevant context.

2. Learning the data manifold (Adversarial Autoencoder). To enforce the distributional constraint, we must be able to identify and penalize out-of-distribution (OOD) actions. We achieve this by training an Adversarial Autoencoder (AAE) on the state-action pairs from the historical data $\mathcal{D}$. The AAE consists of an encoder E and a decoder D. The encoder maps an input state-action pair (s, a) to a latent code $z = E(\sigma(s, a))$, where $\sigma(\cdot)$ is a feature-wise standardizer. The decoder reconstructs the original pair $\hat{a}, \hat{s} = D(z)$.

The AAE is trained with a composite loss function that combines a reconstruction objective with an adversarial regularizer, which forces the latent space to match a prior distribution (e.g., $\mathcal{N}(0, I)$):

$$\mathcal{L}_{\text{AAE}} = \underbrace{\|[\sigma(s, a)] - D(E(\sigma(s, a)))\|_2^2}_{\text{Reconstruction Loss}} + \beta \cdot \underbrace{\text{JS}\left(q(z) \,\|\, p(z)\right)}_{\text{Adversarial Regularizer}}. \tag{8}$$

Once trained, the AAE provides a powerful OOD regularizer. We define the *latent inconsistency penalty* as the reconstruction error:

$$\phi_{\text{lat}}(s,a) = \|\sigma(s,a) - D \circ E(\sigma(s,a))\|_2^2 . \tag{9}$$

A high value for $\phi_{\text{lat}}(s,a)$ indicates that the pair (s,a) lies far from the manifold of the training data.

Offline output

$$\mathcal{M}_{\text{offline}} \triangleq \left\{ E, D, \phi_{\text{lat}}, \hat{Q}_\theta, \sigma(\cdot) \right\} \tag{10}$$

The set $\mathcal{M}_{\text{offline}}$ is exported as a frozen offline artifact, ready to be deployed by the online controller for real-time action selection.

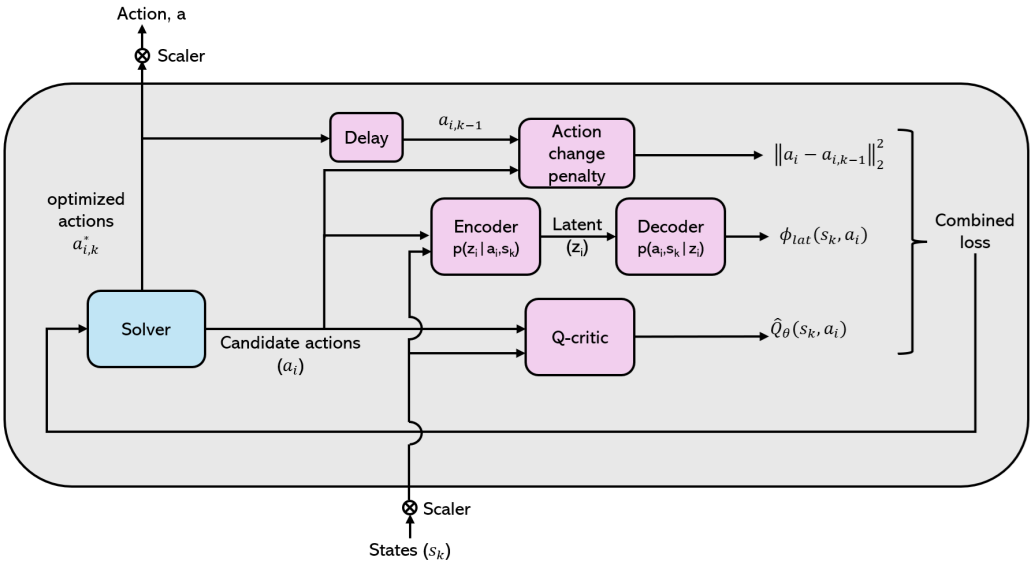

Fig. 2: Block diagram representation of the online elements in the HOFLON-RL algorithm.

2.2.2 Online optimization phase

During online deployment, the trained and frozen models $\{\hat{Q}_\theta, E, D\}$ are used by a real-time optimizer to select the next action at each control step k. The controller receives the current state s_k, computes the error terms e_k and I_k, and then solves the following constrained optimization problem to find the optimal action $a_k^\star$:

$$a_k^\star = \arg\max_{a \in [0,1]^m} \left[\hat{Q}_\theta(s_k, e_k, I_k, a) - \lambda_1 \phi_{\text{lat}}(s_k, a) - \lambda_2 \|a - a_{k-1}\|_2^2 \right]. \tag{11}$$

The objective function balances three critical goals:

- **Value maximization:** The term $\hat{Q}_\theta(\cdot)$ drives the policy to select actions that lead to high long-term returns.
- **Safety/distributional constraint:** The penalty $\phi_{\text{lat}}(\cdot)$ ensures the selected action is "in-distribution," preventing the optimizer from exploiting erroneous regions of the Q-function.
- **Control smoothness:** The term $\|a - a_{k-1}\|_2^2$ penalizes large changes in consecutive actions, ensuring smooth actuator movements.

The weights λ_1 and λ_2 are hyperparameters that tune the trade-off between performance, safety, and smoothness. This optimization problem is solved at each time step using a numerical solver (e.g., Powell's method), as outlined in Algorithm 1 and illustrated in Fig. 2. This formulation translates the rich information learned from offline data into a high-performance, constraint-aware, and stable online control policy.

Algorithm 1: HOFLON-RL

Data: initial observation x_0; previous action a_{-1}; integrator vector $I_0 = \mathbf{0}$
Input: weights (f_1, f_2, f_3); sampling period Δt; tolerance ε; corridor length n_c

for $k = 0, 1, 2, \ldots$ **do**
  $x_k \leftarrow$ **observe** plant;
  $e_k \leftarrow g\big(x_k, x_{\mathrm{sp}}\big)$;
  / derivative-free optimization */*;
  $a_k^\star \leftarrow$ Powell $J(a), [0,1]^m$;
  apply $a_k^\star$; wait Δt and receive x_{k+1};
  if $a_k^\star$ *not clipped* **then**
    $I_{k+1} \leftarrow I_k + e_k\,\Delta t$;
  else
    $I_{k+1} \leftarrow I_k$;
  if $|e_k^{(i)}| < \varepsilon\ \forall i$ *for* n_c *steps* **then**
    $(f_2, f_3) \leftarrow$ `tighten_weights`();

3 Case studies

We assess the proposed HOFLON–RL approach on two challenging, industrially motivated simulation benchmarks that capture common operational hurdles in process industries. Both are formulated as Multiple-Input, Multiple-Output (MIMO) control problems with significant cross-couplings, requiring coordinated multivariable decision-making under constraints.

- **Polymerization reactor start-up.** The first benchmark considers the start-up of an exothermic polymerization CSTR based on the control-oriented model of Maner and Doyle (1997). The control objective is to drive the reactor from an inert initial condition to an operating region following temperature and conversion targets, reflecting the risk of thermal runaway that characterizes many exothermic reaction systems. We emphasize that this benchmark is used to evaluate HOFLON's ability to learn and execute transition policies from offline data; accordingly, the underlying reactor model intentionally adopts simplifying assumptions (e.g., a reduced radical mechanism, neglected gas-phase formation/pressure dynamics, and constant thermophysical properties) to isolate the closed-loop decision-making challenge.
- **Paper-machine grade change.** The second benchmark focuses on economic performance during a product grade change using the multivariable paper-machine model of Kao et al. (2025). The controller must transition multiple quality variables (basis weight, ash content, and moisture) to new specifications while minimising off-spec production and avoiding excessive actuation, representing a canonical multivariable grade-transition problem.

Both benchmarks are released as fully Python implementations, together with curated offline datasets and scripts for offline training and online control, at `https://github.com/AISL-at-Imperial-College-London`. Additional benchmark specifications are provided in the Appendix.

4 Results

In this section, we report the empirical performance of HOFLON-RL on the two benchmark problems introduced earlier. We begin with the polymerization-reactor start-up, detailing the specific network architecture, hyper-parameters and training protocol used for each HOFLON-RL formulation, then present the results that highlight the tracking performance. The second part of the section applies the same evaluation to the paper-machine grade-change problem, allowing a comparison of across the two different MIMO problems. IQL baseline design and the results using the IQL agents are provided for comparison for both case studies.

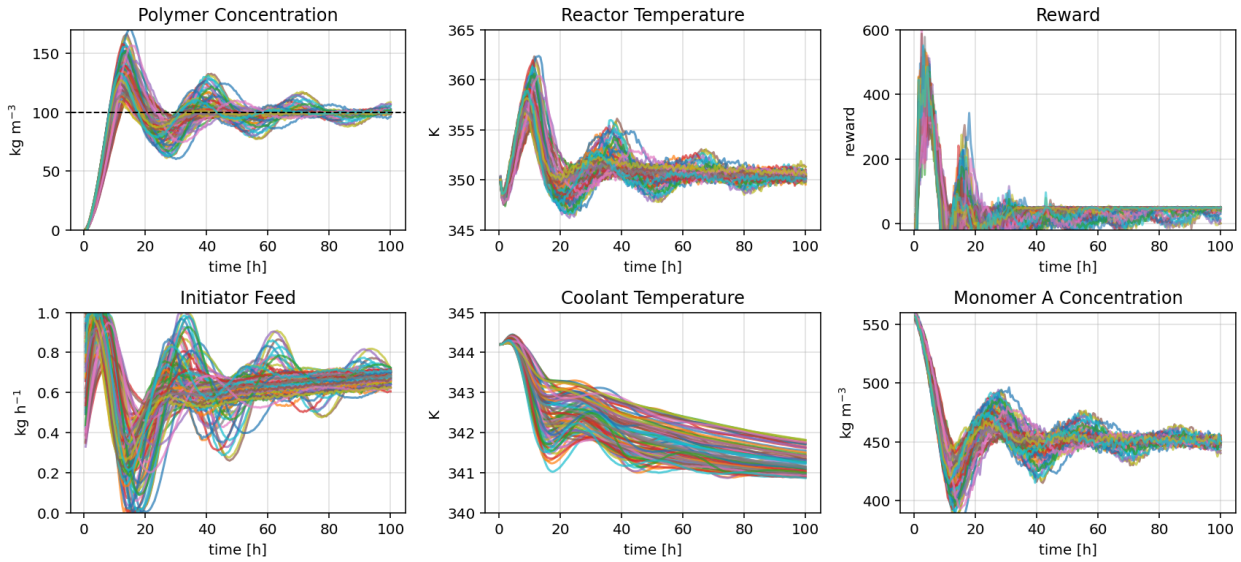

Fig. 3: 100 PI controlled episodes from the polymerization CSTR start-up scenario.

4.1 Polymerization reactor start-up

Fig. 3 depicts 100 simulated episodes of the polymerization-CSTR, each driven by a PI controller whose gains were sampled at random. The trajectories span the full range of closed-loop behavior observed in the start-up scenario: some runs settle the polymer concentration and temperature rapidly at their set-points, whereas others display oscillations or sluggish convergence caused by less favorable tuning. Such variability in the data is important for offline RL because it supplies a rich collection of state–action pairs from both well-regulated and poorly regulated regimes. Exposing the learning agent to this broad operating envelope—even those episodes that verge on unsafe—helps foster policies that are robust and, importantly, safe across diverse process conditions.

4.1.1 HOFLON: offline learning (polymerization)

Offline training consists of two parts: (i) fitting a value function $Q_\theta(s, a)$ on the historical returns and (ii) learning a low-dimensional manifold of feasible state–action pairs with an adversarial auto-encoder (AAE). The resulting artifacts are frozen and used by the online optimizer only for inference.

Table 1: Design parameters for the XGBoost Q-critic

Feature vector size	11 (9 state + 2 action)
Discount factor	$\gamma = 0.90$
Model type	XGBRegressor
Trees / depth	1600 / 8
Learning rate	0.03
ℓ_2 regularisation	$\lambda = 1$
Train/validation split	80 / 20 %
Hold-out R^2	0.96

Q-critic. The action-value function, or Q-critic, is approximated using an XGBoost regressor. The model's purpose is to estimate the expected discounted future return for any given state-action pair. The input to the regressor is a single feature vector formed by concatenating the nine-element state observation $\left[M, I, R, P, T_s, e_P, e_T, i_P, i_T\right]$ with the two-element action vector $[u_I, u_{T_c}]$.

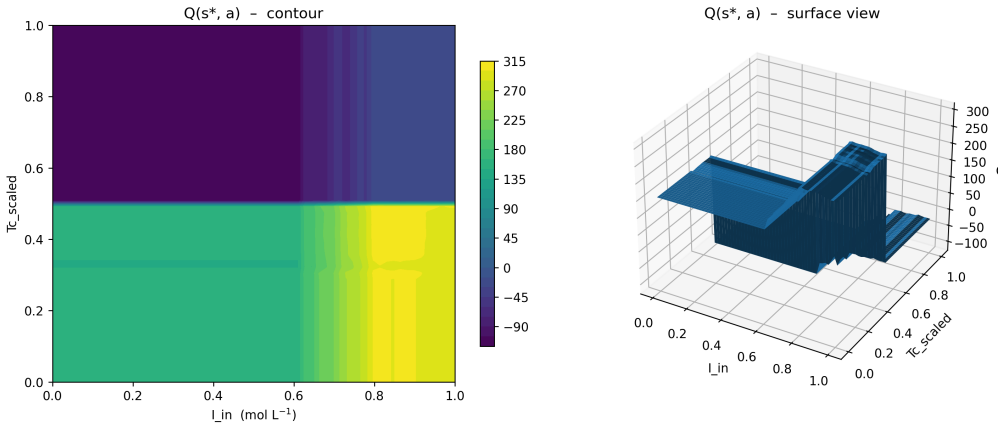

Fig. 4: The Q-critic surface over the action space evaluated at a state observation from the historical dataset.

The key hyperparameters for the model, such as the number of trees and learning rate, are summarized in Table 1. Following training, the model showed strong generalization, achieving a coefficient of determination (R^2) of 0.96 on a 20 % hold-out validation set. A visualization of the learned Q-function is presented in Figure 4. The plot shows the critic's output over the entire action space for a fixed, representative state. The resulting surface exhibits piecewise constant and highly nonlinear features expected from an XGBoost model.

Table 2: Network architecture and training settings for the AAE

Component	**Layer**	**Output Dim.**	**Activation**
Encoder	Input	11	–
	FC + BN	64	LeakyReLU(0.1)
	FC	32	LeakyReLU(0.1)
	FC (latent z)	4	–
Decoder	FC	32	Tanh
	FC	64	Tanh
	FC	11	Tanh
Discriminator	FC	32	LeakyReLU(0.2)
	Dropout	–	$p = 0.3$
	FC	16	LeakyReLU(0.2)
	FC	1	Sigmoid

Hyperparameter	**Value**
Latent prior	$\mathcal{N}(0, I_4)$
Optimiser	Adam ($\alpha = 10^{-5}, \beta_1 = 0.5, \beta_2 = 0.999$)
Batch size	256
Epochs	250 (patience 20)
WGAN Gradient Penalty	10
Test MSE	1.6×10^{-3}

Adversarial auto-encoder. To create a compact and regularized generative manifold of the system data, we employ an Adversarial Autoencoder (AAE). The AAE comprises an encoder, a decoder, and an adversarial discriminator, whose specific network architectures and training settings are detailed in Table 2. All input features are first min-max scaled to the range $[-1, 1]$ to match the decoder's tanh output activation. The discriminator's role is to force the distribution of the four-dimensional latent code from the encoder to

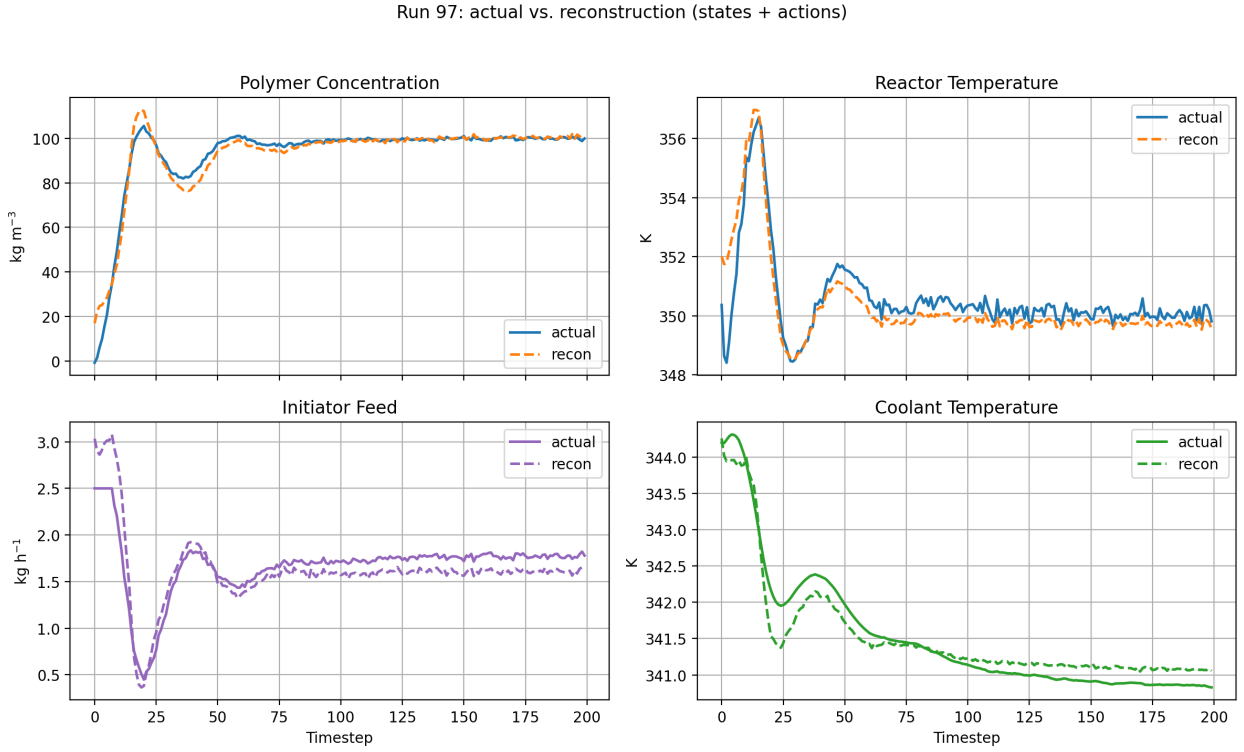

Fig. 5: Reconstruction of a held-out polymerization run using the AAE for the variables and the actions.

match a standard normal prior, $\mathcal{N}(0, I_4)$. This adversarial training disentangles the latent space, making it smooth and continuous—a vital property for effective sampling during online optimization. The trained AAE using the trajectories from the first 85 runs demonstrates high fidelity in reconstructing the data manifold, achieving a low mean squared error (MSE) of 1.6×10^{-3} on the held-out runs. This performance is visually confirmed in Fig. 5, which overlays an original data trajectory with its reconstruction, showing good agreement.

Together, the well-fitted XGBoost critic and the latent AAE manifold model provide the online layer with (i) a dense estimate of long-term value gradients and (ii) a data-supported constraint that penalizes excursions into regions not seen in historical operation.

4.1.2 HOFLON: online optimization (polymerization)

Single–step objective. At each sampling instant the controller solves the box-constrained optimization given in 11 The weights λ_1 and λ_2 are chosen as 0.08 and 0.04 respectively.

Solution of the optimization problem. Because the objective is non-differentiable but cheap to evaluate the following derivative-free solvers are launched sequentially:

1. **Nelder–Mead simplex**. A local simplex that explores a smooth neighborhood extremely quickly. It lacks native bound handling, so its raw optimum is clipped back to $[0, 1]^2$ before J the objective is re-evaluated.

2. **Powell's conjugate-direction search**. Another local pattern search that *does* enforce bounds and often escapes the simplex if Nelder–Mead stalls in a flat valley.

3. **DIRECT global pattern search** Provides a lightweight, deterministic global scan of the unit square.

After the three calls the controller re-computes the objective for the *clipped* candidates and applies whichever action set yields the lowest cost. The simulations were performed on a laptop equipped with an **Intel Core Ultra 9 185H** processor (16 physical cores, 22 hardware threads, base-clock 2.5 GHz). The mean wall-clock per call was 239 ms (median 233 ms, maximum 443 ms), which is comfortably below the sampling period used in the case study. All solvers were warm-started from the previous control moves.

4.1.3 IQL agent training (polymerization)

Value function (expectiles). We first fit an explicit value function $V_\tau(s)$ by expectile regression on Monte-Carlo returns $G_t = \sum_{k\geq 0} \gamma^k r_{t+k}$. Expectiles are obtained via iterative reweighted least squares with asymmetric squared-loss weights $\rho_\tau(\cdot)$; no per-state quantile shortcut is used.

Q-function (temporal-difference target). Given the expectile value function V_τ, we train the action–value regressor $\hat{Q}(s,a)$ by supervised learning on one-step temporal-difference (TD) targets. For each transition (s_t, a_t, r_t, s_{t+1}), the TD target is

$$y_t \;\coloneqq\; r_t \;+\; \gamma\, V_\tau(s_{t+1}),$$

i.e., immediate reward plus the discounted value of the next state. We then fit $\hat{Q}$ on concatenated $[s_t, a_t]$ features by minimizing $\sum_t \big(\hat{Q}(s_t,a_t) - y_t\big)^2$. Thus, y_t serves as a bootstrapped estimate of $Q^\pi(s_t,a_t)$ derived from V_τ, rather than the true Q itself.

Advantages and IQL weights. Advantages are computed as $A_t = \hat{Q}(s_t,a_t) - V_\tau(s_t)$, and the IQL weights as $w_t = \exp(A_t/\beta)$ with standard numerical clipping for stability.

Policy learning. The policy is trained by weighted regression from state to action, one regressor per action dimension, minimizing $\sum_t w_t \|\pi_\phi(s_t) - a_t\|_2^2$. Design parameters for the policy models are summarized in Table 3.

Table 3: Design parameters for the IQL XGBoost policy

Feature vector size	9 (state only)
Model type	XGBRegressor
Trees / depth	1200 / 8
Learning rate	0.03
ℓ_2 regularisation	$\lambda = 1$ (default)
Diagnostics split	80/20 %
Hold-out R^2 (u_I / u_Tc)	0.8988 / 0.8920

4.1.4 Simulation results (polymerization)

We evaluate the proposed HOFLON–RL controller and a trained IQL policy on the polymerization CSTR start-up task. For each controller, we run 100 simulated evaluation episodes under the same operating envelope, initial-condition distribution, and measurement noise used to generate the offline training data. Fig. 6 presents representative trajectories for the evaluation episode with the median cumulative reward across all rollouts, comparing HOFLON, IQL, and the original data-generation strategy. The distribution of start-up tracking performance across the 100 episodes is summarized in Fig. 7. Fig. 8 reports the distribution of cumulative rewards for the same set of episodes. A detailed interpretation of these results is deferred to the Discussion section.

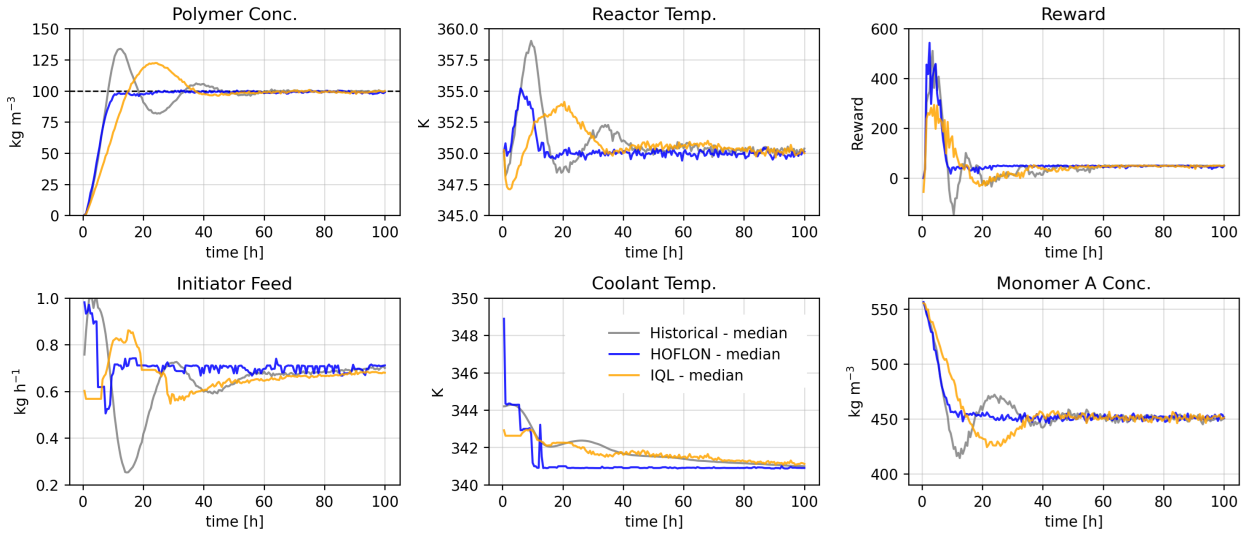

Fig. 6: Start-up trajectories of the polymerization CSTR under HOFLON, IQL, and the original data generation strategy; shown is the evaluation episode with the median cumulative reward across all rollouts.

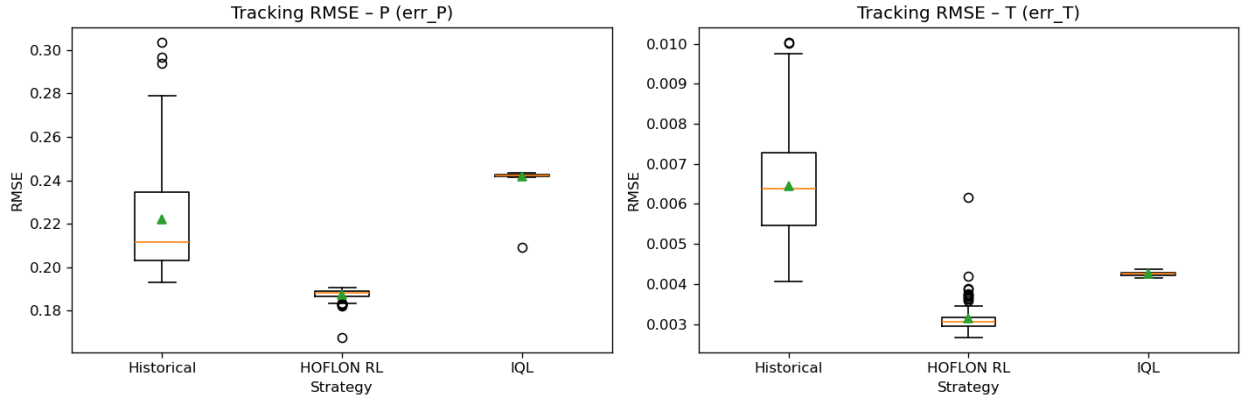

Fig. 7: Start-up tracking performance comparison of HOFLON, IQL, and the original data generation strategy.

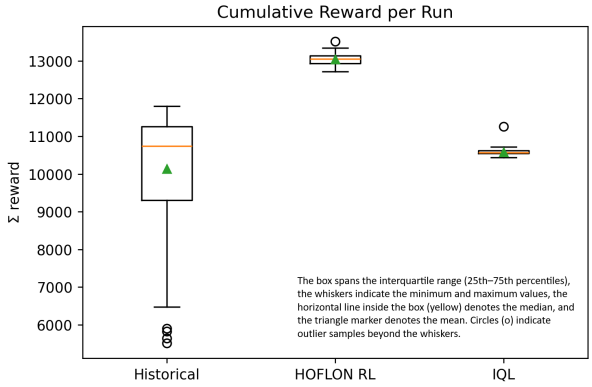

Fig. 8: Start-up cumulative reward comparison of HOFLON, IQL, and the original data generation strategy

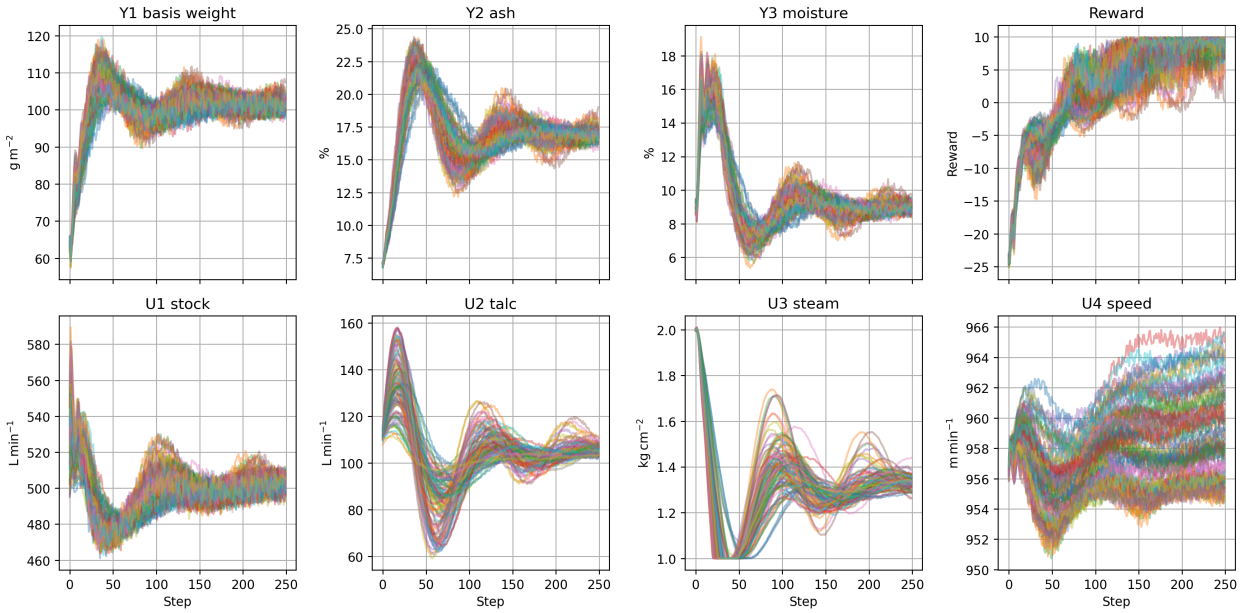

Fig. 9: 100 PI controlled episodes from the paper machine grade change scenario.

4.2 Paper machine grade change

The evaluation protocol for the paper machine grade change problem mirrors the polymerization CSTR study; here we will only note settings specific to the paper machine. Fig. 9 depicts the 100 simulated episodes of the paper machine each driven by a set of PI controllers whose gains were sampled at random.

4.2.1 HOFLON: offline learning (paper machine)

Table 4: Design parameters for the XGBoost Q-critic

Feature vector size	14 (10 state + 4 action)
Discount factor	$\gamma = 0.90$
Model type	XGBRegressor
Trees / depth	1200 / 12
Learning rate	0.02
ℓ_2 regularisation	$\lambda = 1.0$
Train/validation split	90 / 10 %
Hold-out R^2	0.95

Q-critic. The action–value function (Q-critic) is approximated with an XGBoost regressor trained to predict discounted returns ($\gamma = 0.9$). The regressor input is a 14-dimensional feature vector formed by concatenating the ten-element state observation $[Y_1, Y_2, Y_3, e_1, e_2, e_3, \texttt{int1}, \texttt{int2}, \texttt{int3}, \texttt{int4}]$ with the four-element action vector $[U_1, U_2, U_3, U_4]$ (columns in this order). Features are standardized (z-score) using a `StandardScaler` fitted on the training split before both training and inference. The key hyperparameters for the model, such as the number of trees and learning rate, are summarized in Table 4. Following training, the model showed very good generalization, achieving a coefficient of determination (R^2) of 0.95 on a 20 % hold-out validation set. A visualization of the learned Q-function for the paper machine case is not provided in this case due to the higher dimension of the problem but we expect the hypersurface to display similar characteristics as in the first case study.

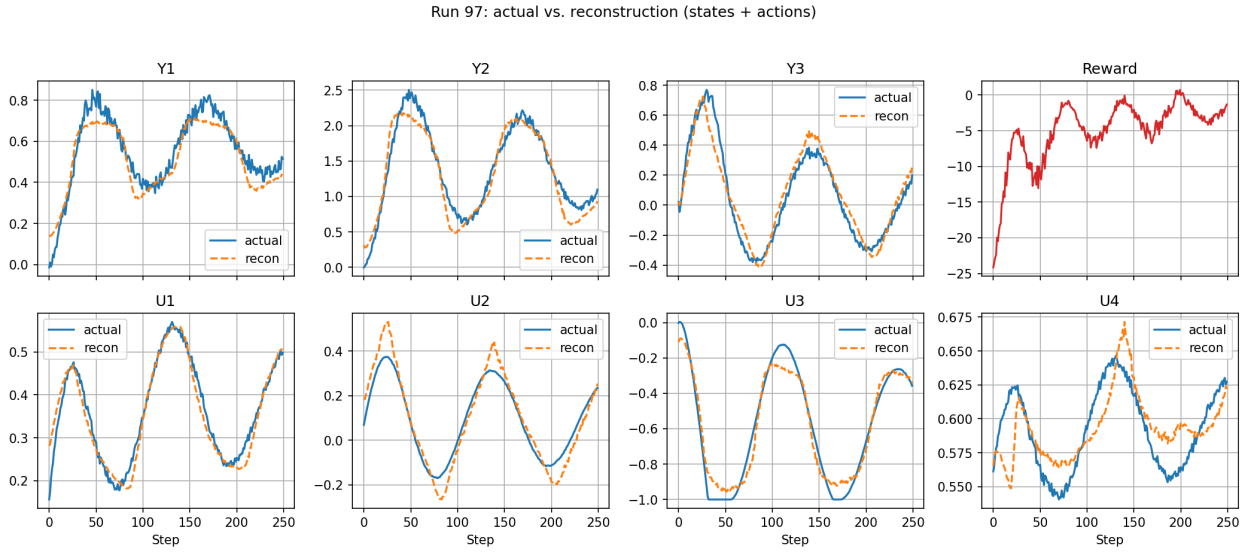

Fig. 10: Reconstruction of a held-out paper machine run using the AEE for the variables and the actions.

Table 5: Network architecture and training settings for the AAE for the paper machine

Component	**Layer**	**Output Dim.**	**Activation**
Encoder	Input	14	–
	FC	64	Tanh
	FC	16	Tanh
	FC (latent z)	4	–
Decoder	FC	16	Tanh
	FC	64	Tanh
	FC	14	–
Discriminator	FC	32	LeakyReLU(0.2)
	FC	16	LeakyReLU(0.2)
	FC	1	Sigmoid

Hyperparameter	**Value**
Epochs	500 (patience 20)
Test MSE	1.7×10^{-2}

All other entries are identical to the previous case study.

Adversarial auto-encoder. As in the previous case study, we use an AAE to obtain a compact generative manifold; here we only note settings specific to the paper-machine problem. The input is a 14-dimensional vector comprising the CVs $[Y_1, Y_2, Y_3]$, error signals $[e_1, e_2, e_3]$, PI integrator states $[\texttt{int1} : \texttt{int4}]$, and MVs $[U_1 : U_4]$, scaled to $[-1, 1]$ using a `MinMaxScaler` fit on the training runs only (IDs 0–94). The encoder is a linear MLP $14 \to 64 \to 16 \to 4$ with tanh activations on hidden layers and a 4-D latent code; the decoder mirrors this as $4 \to 16 \to 64 \to 14$; the discriminator is $4 \to 32 \to 16 \to 1$ with LeakyReLU(0.2) and a sigmoid output. We use Xavier initialisation, MSE reconstruction loss, and a vanilla GAN objective (BCE) for the adversarial term (no gradient penalty), optimized with Adam at 10^{-5} for both autoencoder and discriminator, batch size 256, for 500 epochs. Train/test are split by run ID (train: 0–94; test: 95–99); the saved artifacts are the fitted scaler and the encoder/decoder weights (validation shown on Fig. 10).

4.2.2 HOFLON: online optimization (paper machine)

Single–step objective. As before, at each sampling instant the controller solves the same box-constrained optimization problem given in 11 The weights λ_1 and λ_2 are chosen as 0.45 and 0.0036 respectively.

Solution of the optimization problem. As before, we use a derivative-free portfolio, but for the paper-machine case only two local searches are invoked per step, both warm-started from the previous move and with actions clipped to physical bounds $[u_{\min}, u_{\max}] \subset \mathbb{R}^4$:

1. **Nelder–Mead simplex** (unconstrained; candidate clipped before J is evaluated).
2. **Powell's conjugate-direction** (unconstrained; candidate likewise clipped).

4.2.3 IQL agent training (paper machine)

We retain the pipeline from the previous case study, where the Q-function estimation from HOFLON design step was re-used; the policy model settings are summarized in Table 6.

Table 6: Design parameters for the XGBoost policy (IQL-weighted)

Feature vector size	10 (state only)
Model type	XGBRegressor
Trees / depth	800 / 8
Learning rate	0.05
ℓ_2 regularization	$\lambda = 1$ (default)
Diagnostics split	Train/hold-out by run ID (last 5 runs held out)
Hold-out R^2 ($U_1/U_2/U_3/U_4$)	0.8757 / 0.8280 / 0.9383 / 0.8786

4.2.4 Simulation results (paper machine)

We evaluate the proposed HOFLON–RL controller and a trained IQL policy on the paper machine grade chnage problem. For each controller, we run 100 simulated evaluation episodes under the same operating envelope, initial-condition distribution, and measurement noise used to generate the offline training data. Fig. 11 presents representative trajectories for the evaluation episode with the median cumulative reward across all rollouts, comparing HOFLON, IQL, and the original data-generation strategy. The distribution of start-up tracking performance across the 100 episodes is summarized in Fig. 12. Fig. 13 reports the distribution of cumulative rewards for the same set of episodes. A detailed interpretation of these results is deferred to the Discussion section.

5 Discussion

Across both benchmarks, HOFLON delivers the highest cumulative reward with low dispersion across episodes, indicating that the decision–time optimization produces consistently good sequences. On the *polymerization start-up*, Table 7 shows that HOFLON attains the lowest tracking error on both tracked variables (e_P and e_T) and the highest cumulative reward among all strategies. This suggests that, for this system, the value critic and the manifold regularization are complementary rather than competing: the critic reliably identifies high-return directions within the portion of the data manifold relevant to start-up, and the move penalty suppresses the impulsive corrections that would otherwise excite thermal dynamics. IQL, in contrast, exhibits higher RMSE for both variables and lower return, consistent with a deterministic policy that hews closely to the behavior manifold and forgoes slightly off-manifold action combinations that the critic judges more valuable; the performance of the historical policy lies between the two in both error and reward.

On the *paper-machine grade change* problem, as summarized in Table 7: HOFLON achieves substantially higher return than both the historical controller and IQL while exhibiting an interpretable tracking pattern.

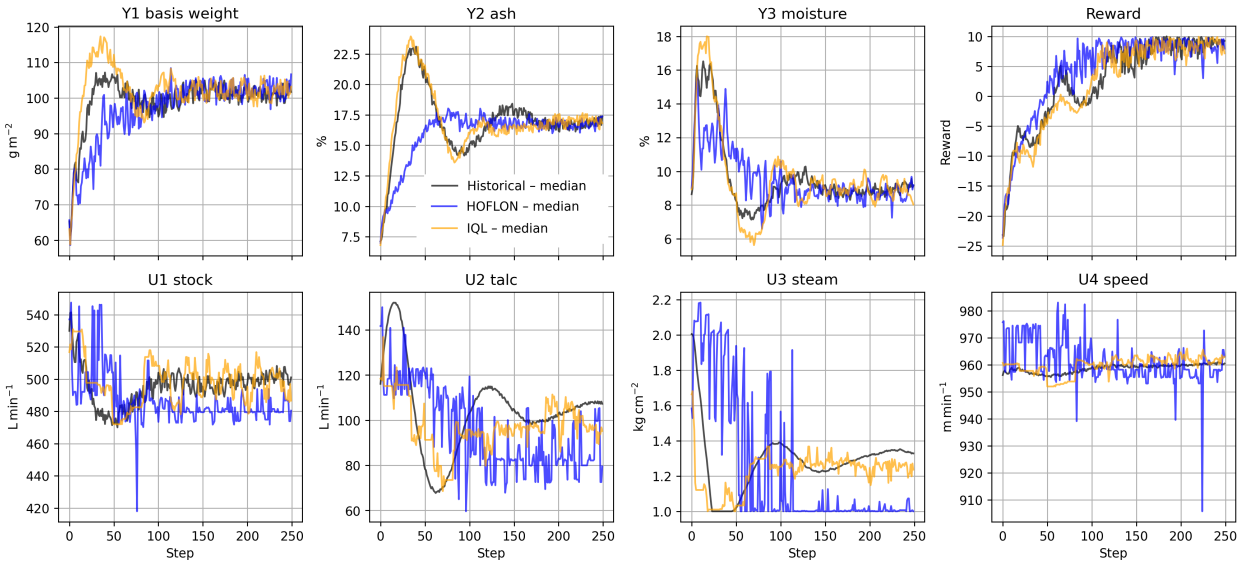

Fig. 11: Paper grade-change trajectories under HOFLON, IQL, and the original data generation strategy; shown is the evaluation episode with the median cumulative reward across all rollouts.

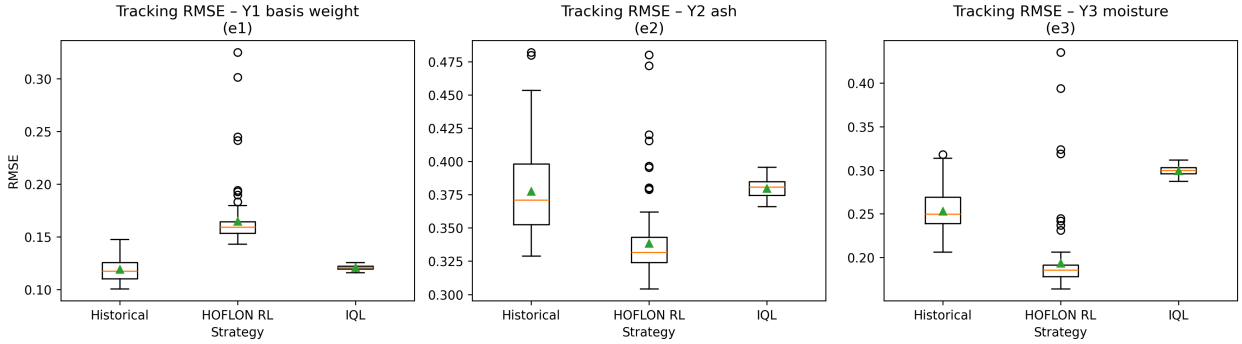

Fig. 12: Paper grade-change tracking performance comparison of HOFLON, IQL, and the original data generation strategy.

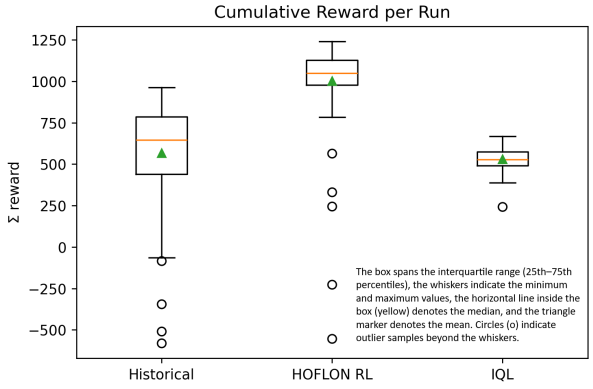

Fig. 13: Paper grade-change cumulative reward comparison of HOFLON, IQL, and the original data generation strategy

Table 7: Summary across both case studies (mean ± std over 100 episodes)

Polymerization start-up			
Strategy	**RMSE e_P**	**RMSE e_T**	**Cumulative reward**
Historical	0.222 ± 0.026	0.006 ± 0.001	10135 ± 1561
IQL	0.242 ± 0.003	0.004 ± 0.000	10583 ± 86
HOFLON RL	$\mathbf{0.187 \pm 0.003}$	$\mathbf{0.003 \pm 0.000}$	$\mathbf{13044 \pm 139}$

Paper-machine grade change				
Strategy	**RMSE $\mathbf{Y_1}$**	**RMSE $\mathbf{Y_2}$**	**RMSE $\mathbf{Y_3}$**	**Cumulative reward**
Historical	$\mathbf{0.119 \pm 0.011}$	0.378 ± 0.034	0.253 ± 0.024	568 ± 313
IQL	0.121 ± 0.002	0.380 ± 0.007	0.300 ± 0.005	531 ± 82
HOFLON RL	0.165 ± 0.026	$\mathbf{0.338 \pm 0.029}$	$\mathbf{0.194 \pm 0.039}$	$\mathbf{1003 \pm 253}$

Ash (Y_2) and moisture (Y_3) RMSEs are reduced relative to both comparators, whereas basis-weight (Y_1) RMSE increases modestly. This trade-off aligns with the objective exposed at action selection: the value term pulls toward trajectories that the critic deems high-return; the data-manifold term (AAE) discourages excursions into poorly supported regions of state–action space; and the move penalty tempers aggressive adjustments that could destabilize the process or over-travel actuators. Given the SIMO pairing (Y_1 driven by both U_1 and U_4), these ingredients can prioritize moisture and ash regulation when sharp basis-weight corrections would erode overall return—mirroring plant economics where transient moisture/ash excursions are often penalized more heavily than brief, small deviations in basis weight. It is also worth noting that, unlike the polymerization case—which exposes internal states to the learner—the paper-machine study trains only on measured outputs, errors, and integrator values, and operates in a larger MIMO setting (four inputs, three outputs). The lack of state measurements and the higher dimensionality make the critic harder to fit and the online objective rougher to optimize, which likely explains the overall weaker tracking levels observed on this task despite the clear reward advantage.

Methodological interpretation Conceptually, HOFLON performs a *regularized greedy policy-improvement step* at each decision point. Pure greed—maximizing a learned critic—can be brittle in offline RL because function approximators may overestimate value in out-of-distribution regions. The AAE manifold therefore acts as a behavior regularizer: it keeps the optimizer near the logged data support unless the critic indicates sufficiently large value gains to justify a controlled deviation. The move penalty adds an orthogonal regularizer that smooths control actions and suppresses myopic jumps. This combination explains why HOFLON realizes higher returns than IQL without sacrificing stability: the method is "greedy where safe, conservative where uncertain." By comparison, IQL trains a deterministic policy via weighted regression toward dataset actions, guided by an expectile value function that stays in-distribution; this yields low-variance behavior but tends to forgo slightly off-manifold action combinations that the critic judges more valuable, resulting in narrower dispersion but systematically lower cumulative reward.

Computational feasibility The online HOFLON controller solves a compact, warm-started derivative-free optimisation problem at each sampling instant, making it suitable for supervisory deployment in industrial control settings. For the action dimensions considered in this work, the resulting wall-clock cost remains comfortably below the corresponding sampling intervals (e.g., tens of seconds to minutes in typical grade-change and start-up supervision), leaving ample time margin on standard industrial compute platforms (industrial PCs or embedded controllers) and, where applicable, enabling execution outside the fastest regulatory loops. While local derivative-free methods applied to a nonconvex objective can be sensitive to initialisation, warm-starting from the previous control move provides continuity from step to step and substantially improves reliability in practice. In addition, the manifold regularisation term discourages large excursions into poorly supported regions of the action space, which empirically reduces erratic search behaviour and promotes smoother optimisation trajectories. For the polymerisation benchmark, we optionally augment the local search portfolio with a lightweight global scan to reduce the likelihood of convergence

to poor local optima; the same option can be enabled for the paper-machine benchmark if future tuning indicates benefit.

Weighting the objective: practical tuning challenges A practical challenge is choosing the relative weights between value maximization, manifold consistency, and move suppression. Even with normalized features and actions, these terms have different scales and interact nonlinearly through the plant dynamics: increasing the manifold penalty can hide local critic overestimation but may also discourage beneficial exploratory moves; strengthening the move penalty improves smoothness and actuator hygiene but can slow convergence; and amplifying the value term can raise return while making the optimizer more sensitive to local roughness in $\hat{Q}$. Here we selected a single set of weights per case study and did not perform fine-grained, channel-wise tuning; the reported performance is therefore a conservative lower bound. Automated schedules or adaptive penalty rules are promising next steps.

Model choices under data and runtime constraints Model choice was deliberately simplified. We adopted XGBoost for the critic and for the IQL policy regressor as a pragmatic bias–variance compromise that trains quickly, offers strong tabular performance, and keeps inference cost modest during online optimization. We experimented with Gaussian-process regression using a 500-point subset (out of $\sim$ 25,000 samples) to keep cubic training cost manageable; the GP produced a smooth surface that was easy to optimize online, but held-out R^2 was poor and closed-loop returns degraded. Lightweight fully connected neural networks underfit and did not perform well; we chose not to scale to larger networks given added training complexity and the risk of sharper nonconvexities in the decision-time objective. Nonetheless, the piecewise-constant nature of boosted trees induces a non-smooth objective; in principle, replacing the ensemble with a smoother regressor that preserves predictive accuracy (e.g., splines, calibrated shallow NNs, or sparse/inducing-point GPs) could ease optimization and improve returns. We view this as a promising avenue for future work.

Per-channel weighting and reward design We did not tune individual term weights per controlled variable. In the paper-machine problem, for example, a CV-specific move penalty or manifold weight could temper the observed trade-off on basis weight without compromising ash and moisture regulation. Such per-channel weighting—and schedules that adapt penalties as the controller approaches target—are natural extensions. Reward design is similarly central. We used a dual *progress-and-proximity* reward: one component rewards directional progress toward the target; the other rewards staying near the target once reached. This structure balanced swift transitions with steady regulation and enabled HOFLON to prioritize moves that both approach and hold the set-point corridor. We did not further tune the relative scaling or shaping; in deployment, start-up or grade-change procedures may warrant finer reward definitions (e.g., asymmetric costs, CV-specific corridors, or time-varying priorities) to encode plant economics and safety envelopes more precisely.

Limitations of the current study Both critic and manifold are trained on finite logs and inevitably carry approximation error; the manifold reduces but does not eliminate the risk that the optimizer exploits local artifacts. The observed increase in Y_1 RMSE on the paper machine is an intentional outcome of the chosen weights and pairing; depending on economics, this balance can be shifted by reweighting the objective or by introducing CV-specific set-point corridors. Results are simulation-based; deployment will require state estimation, actuator rate/travel limits, and online anomaly detection, as well as automated schedules for the value/manifold/move weights to reduce manual tuning effort. The representative datasets for the setting considered in this work are trajectories generated under feedback, either by human operators (human-in-the-loop) or by existing regulatory controllers (controller-in-the-loop). In our simulation study, we generate offline data using baseline feedback controllers as reproducible surrogates for such behavior policies. To emulate the fact that historical logs may reflect *multiple* operating styles (e.g., different operators, shifts, or controller configurations), we deliberately vary the controller settings across rollouts. We note as a limitation that the present study does not include genuine human-in-the-loop operating data; evaluating HOFLON on industrial transition logs and/or operator-driven simulator rollouts is an important direction for future work.

Toward a generalized, transition-agnostic HOFLON The results suggest a path toward an operations-wide agent that can handle arbitrary start-ups and grade changes without retraining. A generalized critic can embed the desired transition directly into its input—set-point paths, ramp rates, and quality envelopes—so a single agent plans many transitions. To preserve fast decision-time optimization, convex-in-action architectures such as input-convex neural networks (ICNNs) are attractive: nonlinear in state yet convex in action, reducing runtime search to a single convex program. The representation cost of convexity should be weighed against alternatives (monotone networks, quadratic forms, Gaussian-process hybrids) to find the best accuracy–solve-time trade-off. Conditioning should also be explicit in the generative component: our current AAE imposes a global manifold over (s, a), whereas conditional latent-variable models (e.g., CVAEs) can model the *conditional* action distribution given state and transition intent—potentially improving optimizer conditioning. Finally, testing without plant models or simulators motivates a dual track: offline back-testing on archived transitions to quantify data sufficiency, coverage gaps, and sensor corruption sensitivity; and live shadow trials that log the agent's proposed moves in parallel with the existing control system. Metrics such as action mismatch, predicted-versus-realized reward, and distance to safety envelopes can be monitored to decide whether recommendations are acceptably close, systematically biased, or occasionally unsafe; robust thresholds and statistical tests for "far-off" suggestions remain an open research problem.

Broader applicability and extensions Although this paper focuses on reactor start-up and paper-machine grade change, the HOFLON framework is not tied to a particular process class. It is most naturally suited to *episodic transition operations* with a clear start and end point, where performance is evaluated over a finite horizon and where historical trajectories are available but direct exploration is undesirable. Beyond the benchmarks studied here, this includes recipe-driven operations, particularly in (semi-)batch processing where repeated campaigns generate rich logs and where transition objectives are naturally stage-wise. It also includes energy-system start-up and shut-down sequences or load changes, such as bringing a boiler or gas turbine online, where equipment protection and operational constraints impose strict limits and where smooth actuation is critical.

A practical advantage of HOFLON is that action selection is formulated as an explicit optimization problem with separable and tunable terms (critic value, data-support/manifold penalty, and move-suppression), which lends itself naturally to operator-facing explanations. At runtime, the controller can report a decomposition of the selected action's score into these components, enabling a simple rationale such as: "this move is preferred because it improves long-horizon value while remaining well supported by historical operation and avoiding aggressive actuation." When the optimization identifies multiple competitive candidates (e.g., solutions with similar objective values), the system can present an explicit trade-off to the operator—for instance, choosing between (i) a higher predicted value option, (ii) an option closer to the learned data manifold, or (iii) a smoother action sequence—together with the corresponding quantitative scores. More detailed visualizations are also possible, such as projecting the learned manifold or latent representations to two dimensions (e.g., via t-SNE) to show where the current state–action pair lies relative to historical trajectories, monitoring the reconstruction/error score over time as an "out-of-distribution" indicator, or displaying the evolving control trajectory alongside representative nearest-neighbor historical rollouts. These interfaces support transparency and facilitate incremental deployment (e.g., advisory mode) in industrial transition operations.

6 Conclusions

We presented HOFLON, a hybrid framework that couples an offline-learned value critic and a data manifold with online, solver-based action selection. On two industrially motivated benchmarks—a polymerization start-up and a paper-machine grade change—HOFLON consistently achieved the highest cumulative reward among all policies tested and, in the paper machine, substantially improved ash and moisture tracking at the cost of a modest increase in basis-weight RMSE. These results underscore three advantages of the approach: (i) explicit, tuneable trade-offs among performance, safety, and move smoothness at decision time; (ii) reduced OOD risk through manifold regularization; and (iii) real-time feasibility with simple derivative-free searches. The framework is broadly applicable to transition operations where accurate first-principles models are unavailable but historical logs exist. Future work will target automatic weight scheduling; compare

smoothed/ensembled critics; refine reward shaping; develop OOD and safety diagnostics using the latent penalty; and pursue the development of transition-agnostic critics and conditional manifolds (e.g., ICNNs, CVAEs) to enhance consistent and fast decision-time optimization across start-ups and grade changes.

References

Brandon Amos, Lei Xu, and J. Zico Kolter. Input convex neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 146–155, 2017.

Michael Baldea, Apostolos T Georgiou, Bhushan Gopaluni, Mehmet Mercangöz, Constantinos C Pantelides, Kiran Sheth, Victor M Zavala, and Christos Georgakis. From automated to autonomous process operations. *Computers & Chemical Engineering*, 196:109064, 2025.

Selwa BenAmor, Francis J Doyle III, and Randall McFarlane. Polymer grade transition control using advanced real-time optimization software. *Journal of Process Control*, 14(4):349–364, 2004.

Dominique Bonvin, Levente Bodizs, and Bala Srinivasan. Optimal grade transition for polyethylene reactors via nco tracking. *Chemical Engineering Research and Design*, 83(6):692–697, 2005.

D.A. Bristow, M. Tharayil, and A.G. Alleyne. A survey of iterative learning control. *IEEE Control Systems Magazine*, 26(3):96–114, 2006. doi: 10.1109/MCS.2006.1636313.

C Chang, H Lee, and Vincentius Surya Kurnia Adi. *Process Plant Operating Procedures*. Springer, 2021.

Shi-Chang Chang, Chun-Yung Chang, Hao-Yeh Lee, and I-Lung Chien. Grade transition optimization by using gated recurrent unit neural network for styrene-acrylonitrile copolymer process. In *Computer Aided Chemical Engineering*, volume 49, pages 1723–1728. Elsevier, 2022.

Xuyang Chen, Keyu Yan, and Lin Zhao. Taming ood actions for offline reinforcement learning: An advantage-based approach. *arXiv preprint arXiv:2505.05126*, 2025.

Hyun-Kyu Choi, Sang Hwan Son, and Joseph Sang-Il Kwon. Inferential model predictive control of continuous pulping under grade transition. *Industrial & Engineering Chemistry Research*, 60(9):3699–3710, 2021.

Stephen Chu. Simplified automatic nonlinear grade change control for paper machines. In *PaperCon 2019 Proceedings*, pages 570–579, Indianapolis, IN, USA, 2019. TAPPI Press.

Prodromos Daoutidis, Jay H Lee, Srinivas Rangarajan, Leo Chiang, Bhushan Gopaluni, Artur M Schweidtmann, Iiro Harjunkoski, Mehmet Mercangöz, Ali Mesbah, Fani Boukouvala, et al. Machine learning in process systems engineering: Challenges and opportunities. *Computers & Chemical Engineering*, 181: 108523, 2024.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, pages 2052–2062, 2019.

Shixiang Gu, Ethan Holly, Timothy Lillicrap, and Sergey Levine. Continuous deep q-learning with normalized advantage functions. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 1995–2003, 2016.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 1352–1361, 2017.

Hossein Mostafaei, Teemu Ikonen, Jason Kramb, Tewodros Deneke, Keijo Heljanko, and Iiro Harjunkoski. Data-driven approach to grade change scheduling optimization in a paper machine. *Industrial & Engineering Chemistry Research*, 59(17):8281–8294, 2020. doi: 10.1021/acs.iecr.9b06907.

H. Ihalainen and R. Ritala. Optimal grade changes. In *Proceedings of Control Systems '96*, pages 213–216, Halifax, Canada, 1996.

Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning (ICML)*, 2022. arXiv:2205.09991.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, and Sergey Levine. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018.

Bingyi Kang, Xiao Ma, Chao Du, Tianyu Pang, and Shuicheng Yan. Efficient diffusion policies for offline reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Heng-Shan Kao, Ming-Wei Chen, Hao-Yeh Lee, Pan-Hsin Wu, Cheng-Liang Chen, Jeffrey D Ward, and I-Lung Chien. Control of melt index in an industrial ethylene-vinyl acetate process using a recurrent neural network soft sensor and multiple virtual control strategies. *Journal of the Taiwan Institute of Chemical Engineers*, 173:106154, 2025.

Donghyeon Ki, JunHyeok Oh, Seong-Woong Shim, and Byung-Jun Lee. Prior-guided diffusion planning for offline reinforcement learning. *arXiv preprint arXiv:2505.10881*, 2025.

Jun-Seok Ko, Yeong-Koo Yeo, Seong-Mun Ha, Jung-Woo Lim, Du-Seok Ko, and Hong Kang. Modeling of grade change operations in paper mills. *Journal of Korea Technical Association of The Pulp and Paper Industry*, 35(5):46–52, 2003.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline Reinforcement Learning with Implicit Q-Learning, 2021. URL `https://arxiv.org/abs/2110.06169`.

Klaus Krüger, Rüdiger Franke, and Manfred Rode. Optimization of boiler start-up using a nonlinear boiler model and hard constraints. *Energy*, 29(12-15):2239–2251, 2004.

Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in neural information processing systems*, 32, 2019.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-Learning for Offline Reinforcement Learning, 2020. URL `https://arxiv.org/abs/2006.04779`.

Per-Ola Larsson, Johan Åkesson, Niclas Carlsson, and Niklas Andersson. Model-based optimization of economical grade changes for the borealis borstar® polyethylene plant. *Computers & chemical engineering*, 46:153–166, 2012.

Kwang S Lee and Jay H Lee. Iterative learning control-based batch process control technique for integrated control of end product properties and transient profiles of process variables. *Journal of Process Control*, 13(7):607–621, 2003.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *4th International Conference on Learning Representations (ICLR)*, 2016. URL `https://arxiv.org/abs/1509.02971`.

Bryon R. Maner and Francis J. Doyle. Polymerization reactor control using autoregressive-plus Volterra-based MPC. *AIChE Journal*, 43(7):1763–1784, 1997. doi: https://doi.org/10.1002/aic.690430713. URL `https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/aic.690430713`.

Liyuan Mao, Haoran Xu, Xianyuan Zhan, Weinan Zhang, and Amy Zhang. Diffusion-dice: In-sample diffusion guidance for offline reinforcement learning. *arXiv preprint arXiv:2407.20109*, 2024.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Hossein Mostafaei, Teemu Ikonen, Jason Kramb, Tewodros Deneke, Keijo Heljanko, and Iiro Harjunkoski. Data-driven approach to grade change scheduling optimization in a paper machine. *Industrial & Engineering Chemistry Research*, 59(17):8281–8294, 2020.

A Papagrigoraki and U Leberle. Best available technique (bat) conclusions for the production of pulp, paper and board. In *Cepi conference) Retrieved from https://www. cepi. org/best-available-technique-bat-conclusions-for-the-production-of-pulp-paper-and-board*, 2015.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-Weighted Regression: Simple and Scalable Off-Policy Reinforcement Learning, 2019. URL `https://arxiv.org/abs/1910.00177`.

Adrian Prata, Jan Oldenburg, Andreas Kroll, and Wolfgang Marquardt. Integrated scheduling and dynamic optimization of grade transitions for a continuous polymerization reactor. *Computers & Chemical Engineering*, 32(3):463–476, 2008.

Rafael Figueiredo Prudencio, Marcos ROA Maximo, and Esther Luna Colombini. A survey on offline reinforcement learning: Taxonomy, review, and open problems. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

Eunhyeok Ryu, Ying Ashitani, Amirhossein Kazemi, and Pascal Poupart. Continuous action q-learning. In *International Conference on Learning Representations (ICLR)*, 2020. URL `https://openreview.net/forum?id=Byx6HkBKDB`.

Richard S. Sutton and Andrew Barto. Reinforcement learning: an introduction, 2018.

S. C. Chen. Modeling of paper machines for control: Theory and practice. *Pulp & Paper Canada*, 96(1): T17–T21, 1995.

Timothy F. Murphy and Shih-Chin Chen. Transition control of paper-making processes: Paper grade change. In *Proceedings of the IEEE International Conference on Control Applications*, pages 1278–1283, Kohala Coast, Hawaii, 1999.

Hado van Hasselt, Arthur Guez, and David Silver. Deep Reinforcement Learning with Double Q-learning, 2015. URL `https://arxiv.org/abs/1509.06461`.

Vipetec Ltd. Results in practice – developing and monitoring grade changes at a paper mill. `https://vipetec.fi/results-in-practice-developing-and-monitoring-grade-changes-at-a-paper-mill/`, 2025. Company case study, accessed 26 July 2025.

Zhendong Wang, Jonathan J. Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2208.06193.

Yifan Wu, George Tucker, and Ofir Nachum. Behavior Regularized Offline Reinforcement Learning, 2019. URL `https://arxiv.org/abs/1911.11361`.

Yeong-Koo Yeo, Jong Ho Park, See-Han Park, and Changman Sohn. Model algorithmic control of grade change operations in paper mills. *Korean Journal of Chemical Engineering*, 22(3):339–344, 2005.

Xinyue Yuan, Jia Li, Changhe Xu, and Zhijun Li. Milp-based Q-learning for constrained supply-chain inventory management. *arXiv preprint arXiv:2305.00001*, 2023.

Wenxuan Zhou, Sujay Bajracharya, and David Held. Plas: Latent action space for offline reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2020.

Appendix

This appendix provides the full benchmark specifications used in the numerical studies to support reproducibility. We document the simulator models and assumptions, parameter values, variable scaling and normalisation, reward and constraint definitions, and the data-generation procedures used to construct the offline datasets (including baseline controller structures, gain ranges, and any excitation settings). The appendix is organised into two parts: (i) the polymerization reactor start-up benchmark and (ii) the paper-machine grade-change benchmark.

A Polymerization reactor start-up

This case study uses the suspended, exothermic CSTR model introduced by Maner and Doyle (1997) to benchmark the HOFLON–RL algorithm on a realistic *start-up* operation. The objective is to drive the polymer concentration C_P from 0 to $100\,\mathrm{kg\,m^{-3}}$ while keeping the reactor temperature near 350 K in the presence of strong heat release and risk of thermal runaway.

A.1 Polymerization reactor model

A schematic of the CSTR is shown in Fig.14. The CSTR is fed with a solvent (S) containing monomer species (M) and an initiator species (I) at temperature T_{in}. In the CSTR, the monomer and initiator react to form a polymer product (P) via a suspended lumped radical (R) mechanism. The CSTR outlet flowrate is assumed to be equal to the total inlet flowrate, maintaining a constant liquid filled reactor volume (1 m^3). Perfect mixing is assumed along with constant fluid density (1000 $\mathrm{kg/m}^3$). Any gas phase effects and changes in viscosity due to polymer build-up are neglected.

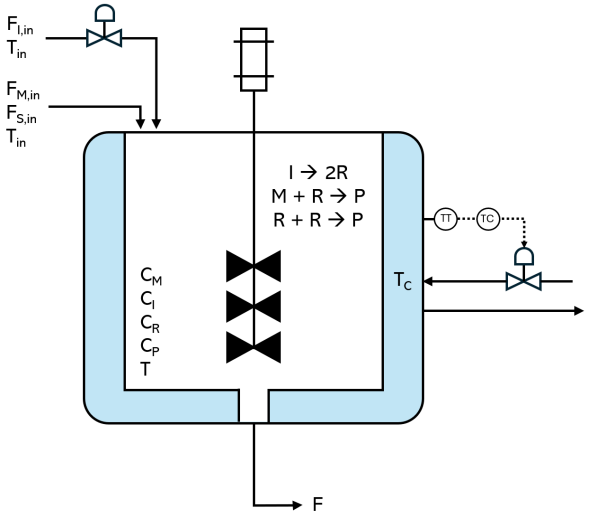

Fig. 14: Schematic of the polymerization CSTR environment. Control actions include the initiator feed rate ($F_{\mathrm{M,in}}$) and cooling jacket temperature T_C. The reactor states include monomer concentration (C_M), initiator concentration (C_I), radical concentration (C_R), polymer concentration (C_P), and reactor temperature (T), where the latter two are also the controlled variables.

The reaction pathway includes thermal auto-decomposition of the initiator to generate radicals (12), propagation of monomer with radicals to form polymer chains (13), and bimolecular termination of radicals (14). In this mechanism, all radical species (including free radical and propagating chain radicals) are represented with a single concentration variable for the lumped radicals, C_R. This approximation simplifies the kinetic model by avoiding the need to track individual chain lengths. The reaction set (12–14) describes the chemical reactions.

$$\text{Initiation:} \quad \mathrm{I} \xrightarrow{\mathrm{k}_i} 2\,\mathrm{R}\cdot \tag{12}$$

$$\text{Propagation:} \quad \mathrm{M} + \mathrm{R}\cdot \xrightarrow{\mathrm{k}_p} \mathrm{P}\cdot \tag{13}$$

$$\text{Termination:} \quad \mathrm{R}\cdot + \mathrm{R}\cdot \xrightarrow{\mathrm{k}_t} \mathrm{P} \tag{14}$$

Reaction kinetics are modeled using temperature-dependent Arrhenius expressions with constant pre-exponential factors and activation energies given in Table 8. All reactions are assumed to be kinetically controlled with no mass transfer limitations.

$$k = A \exp\left(-\frac{E}{RT}\right) \tag{15}$$

The reaction rates are defined as follows:

$$\text{Initiation:} \quad r_i = k_i C_I \tag{16}$$

$$\text{Propagation:} \quad r_p = k_p C_M C_R \tag{17}$$

$$\text{Termination:} \quad r_t = k_t C_R^2 \tag{18}$$

A.2 Dynamics and numerical integration

The reactor dynamics are governed by a system of coupled ordinary differential equations (19–23) using the parameters in Table 8. Heat effects are explicitly modeled, with an exothermic heat of reaction and thermal exchange with a cooling jacket. These are solved at each simulation step using SciPy's solveivp function with a stiff solver (BDF). The integration time step is set to 30 minutes to reflect typical control intervals for large-scale batch or semi-batch reactors. The simulation is terminated after 100 hours of operation unless otherwise specified.

$$\frac{dC_M}{dt} = \frac{F}{V}\left(C_M^{\text{in}} - C_M\right) - r_p \tag{19}$$

$$\frac{dC_I}{dt} = \frac{F}{V}\left(C_I^{\text{in}} - C_I\right) - r_i \tag{20}$$

$$\frac{dC_R}{dt} = \frac{F}{V}\left(C_R^{\text{in}} - C_R\right) + 2r_i - 2r_t \tag{21}$$

$$\frac{dC_P}{dt} = \frac{F}{V}\left(C_P^{\text{in}} - C_P\right) + r_p \tag{22}$$

$$\frac{dT}{dt} = \frac{F}{V}(T_f - T) - \frac{UA(T - T_c)}{\rho C_p V} - \frac{r_p \Delta H_{\text{rxn}}}{\rho C_p} \tag{23}$$

Table 8 gives the parameters used to construct the polymerization reactor environment. The reaction is inspired by the polymerization of vinyl acetate in benzene as presented in (Maner and Doyle, 1997), with some simplifications and rounding of parameters.

A.3 RL observation and action spaces

State scaling. Concentrations are divided by 100 ($\mathrm{kg\,m^{-3}}$) and temperature by 350 K, yielding dimensionless variables in roughly the range $[0, 2]$: $P_s = C_P/100$ and $T_s = T/350$.

Actions. The agent selects two dimensionless inputs in $[0, 1]$:

$$u_I = \frac{f_I - f_{I,\min}}{f_{I,\max} - f_{I,\min}}, \qquad u_{T_c} = \frac{T_c - T_{c,\min}}{T_{c,\max} - T_{c,\min}},$$

with $f_I \in [0, 2.5]$ $\mathrm{kg\,h^{-1}}$ and $T_c \in [280, 400]$ K.

Table 8: Summary of reaction / reactor parameters used in the PolyCSTR environment

Category	Parameter	Value
Physical Constants	Gas constant R	8.314 J/mol/K
	Density ρ	1000 kg/m^3
	Heat capacity C_p	2000 J/kg/K
	Heat transfer coefficient U	500 J/s/K
	Heat of reaction ΔH_{rxn}	-100,000 J/mol
Reactor Configuration	Volume V	1.0 m^3
	Feed solvent rate F_S	80 kg/h
	Feed monomer rate F_M	100 kg/h
	Feed temperature T_f	350 K
Kinetics	A_{init}	1×10^9 1/s
	E_{init}	125,000 J/mol
	A_{prop}	4×10^4 m^3/mol/s
	E_{prop}	25,000 J/mol
	A_{term}	1×10^6 m^3/mol/s
	E_{term}	15,000 J/mol
Control Limits	Initiator feed rate	[0.0, 2.5] kg/h
	Coolant temperature	[300, 355] K
Simulation Settings	Time step	1800 s
	Max simulation time	100 hours

Observations. The agent receives the five scaled states with Gaussian measurement noise ($1\,\sigma$ standard deviations $\sigma = [0.02, 0, 0, 0.005, 0.00057]$).

A.4 Data-generation methodology

Offline datasets are produced by a two-loop PI controller operating in the *scaled* domain.

- **Controller gains.** Base gains $\big(K_{p,P}, K_{i,P}, K_{p,T}, K_{i,T}\big) = \big(1.04,\, 8 \times 10^{-5},\, 0.012,\, 3.8 \times 10^{-5}\big)$ are multiplied by independent random factors sampled uniformly from $[0.3, 1.0]$ at the start of each run.
- **Anti-wind-up.** Integral terms are frozen whenever a manipulated variable saturates.
- **Scheduling.** Each run simulates 100 h with a 30 min control interval, yielding 200 logged points. We generate 100 statistically independent runs, giving 20 000 transitions in the final dataset.
- **Measurement noise.** Additive Gaussian noise with the standard deviations given in Section A.3 is added to the state vector before it is fed to the controller and recorded in the dataset.

Pairing. A diagonal two-loop PI structure is used: the polymer-concentration error e_P actuates the initiator feed u_I, and the temperature error e_T actuates the coolant set-point u_{T_c}; no cross terms are applied.

A.5 Reward function

Let P_s and T_s denote the *scaled* polymer concentration and temperature, with set-points $P^{\text{sp}} = 1$ and $T_s^{\text{sp}} = 1$. Define the instantaneous squared error

$$e(k) = \big(P_s(k) - P^{\text{sp}}\big)^2 + \big(T_s(k) - T_s^{\text{sp}}\big)^2$$

and the improvement $\Delta e(k) = e(k-1) - e(k)$. For actions $u(k) = [u_I(k), u_{T_c}(k)]$ and step changes $\Delta u = u(k) - u(k-1)$, the reward has the form:

$$r(k) = \alpha\Big[-e(k) + \exp\Big(-\tfrac{e(k)}{\sigma^2}\Big)\Big] + \Delta e(k) - \lambda_1\,\Delta u_I^2 - \lambda_2\,\Delta u_{T_c}^2, \tag{24}$$

with $\sigma = 0.05$. The first bracketed term rewards proximity to the set-point, the second rewards progress, and the last term discourages aggressive control moves.

A.6 Logged dataset

At each 30-minute interval, the driver appends one row containing: identifiers (`run_id`, time (min)); scaled states M, I, R, P, T_s; scaled errors `err_P`, `err_T`, `int_err_P`, `int_err_T`; scaled MVs `u_I`, `u_Tc`; and the reward.

A.7 Control scenario

Only the **start-up** scenario is considered:

- *Warm start.* The reactor initially contains solvent and monomer at $T = 350\,\mathrm{K}$ with *no* initiator ($u_I = 0$), hence $P_s(0) = 0$. The task is to reach $P_s = 1$ and maintain $T_s = 1$ without triggering thermal runaway.

The combination of strong exothermic behavior, nonlinear kinetics, measurement noise and randomly perturbed PID loops results in a dataset that contains both cautious and aggressive trajectories—making it an informative benchmark for offline policy learning.

B Paper machine grade change

This case study examines the dynamic modeling of a grade change operation in a paper machine, based on the work by Ko et al. (2003). A grade change involves transitioning the production process from one paper specification to another, a complex procedure that can lead to significant amounts of off-specification product if not carried out precisely. The objective is to minimize this transition time and maintain product quality.

The physical system encompasses the process from the headbox through to the main dryer section and is illustrated in Fig.15. The key process variables are:

- **Manipulated Variables (Inputs or actions):** Stock flow, filler (clay) flow, steam pressure, and machine speed.
- **Controlled Variables (Outputs or observations):** Basis weight, ash content, and moisture content.

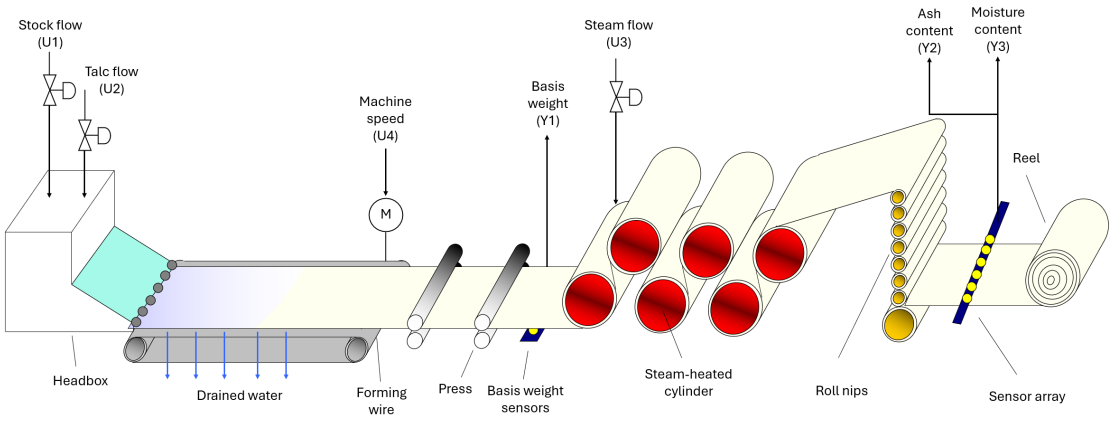

Fig. 15: Schematic of the paper machine indicating the manipulated variables (or actions) and the observed process outputs.

In Ko et al. (2003), closed-loop system identification is used to obtain a discrete-time, linear time-invariant (LTI) state-space model. The model is defined by the state-space matrices A, B, C, and the direct feedthrough matrix D, which is a zero matrix as is common for processes with inherent delays between inputs and outputs.

The state-space model is given by

$$x(k+1) = A\,x(k) + B\,u(k), \tag{25}$$

$$y(k) = C\,x(k) + D\,u(k). \tag{26}$$

Here, $x(k)$ is the state vector, $u(k)$ is the vector of manipulated-variable deviations, and $y(k)$ is the vector of controlled-variable deviations.

The identified matrices are:

State matrix A

$$A = \begin{bmatrix} 0.676 & 0.349 & -0.153 & -2.462 & -2.182 & 0.735 & -0.477 & -1.326 & 0.201 & -0.199 \\ -1.309 & 2.494 & -0.601 & -10.70 & -3.071 & -0.850 & -0.803 & -3.458 & 0.641 & -0.372 \\ 0.941 & -1.215 & 1.472 & 9.160 & 4.575 & -3.024 & 1.526 & 4.882 & -1.176 & 0.902 \\ -0.305 & 0.362 & -0.132 & -1.566 & -0.493 & -0.145 & -0.079 & -0.553 & 0.151 & -0.025 \\ -0.043 & 0.077 & -0.029 & -0.612 & 0.619 & 0.449 & 0.164 & -0.017 & -0.098 & 0.035 \\ -0.318 & 0.358 & -0.108 & -2.318 & 0.066 & 0.193 & -0.523 & -0.131 & 0.146 & 0.064 \\ -0.027 & 0.071 & -0.021 & -0.491 & 0.052 & 0.398 & 0.542 & -0.111 & 0.434 & -0.060 \\ 0.179 & -0.203 & 0.024 & 1.140 & -0.029 & -0.006 & -0.046 & -0.143 & 0.162 & 0.023 \\ 0.437 & -0.511 & 0.161 & 3.481 & 0.593 & 0.311 & -0.526 & -0.215 & 0.584 & 0.220 \\ 0.420 & -0.519 & 0.174 & 3.604 & 0.577 & 0.241 & 0.102 & 0.272 & -0.462 & 0.354 \end{bmatrix}$$

Input matrix B

$$B = \begin{bmatrix} -5.15 \times 10^{-3} & -1.977 \times 10^{-2} & 7.755 \times 10^{-1} & -1.381 \times 10^{-2} \\ -1.938 \times 10^{-2} & -5.644 \times 10^{-2} & 3.630 & -5.918 \times 10^{-2} \\ -1.642 \times 10^{-2} & 3.865 \times 10^{-2} & -5.436 \times 10^{-1} & 5.292 \times 10^{-2} \\ -2.95 \times 10^{-3} & -7.79 \times 10^{-3} & 4.429 \times 10^{-1} & -1.406 \times 10^{-2} \\ 7.13 \times 10^{-3} & -8.50 \times 10^{-4} & -6.990 \times 10^{-1} & -4.00 \times 10^{-3} \\ -1.25 \times 10^{-2} & 3.40 \times 10^{-4} & 8.226 \times 10^{-1} & -1.156 \times 10^{-2} \\ 5.65 \times 10^{-3} & 1.84 \times 10^{-3} & -2.227 \times 10^{-1} & -3.76 \times 10^{-3} \\ 7.23 \times 10^{-3} & -1.536 \times 10^{-2} & 9.547 \times 10^{-1} & 4.46 \times 10^{-3} \\ 6.13 \times 10^{-3} & 8.90 \times 10^{-4} & 2.952 \times 10^{-1} & 1.798 \times 10^{-2} \\ 1.09 \times 10^{-3} & 4.58 \times 10^{-3} & 4.471 \times 10^{-2} & 1.986 \times 10^{-2} \end{bmatrix}$$

Output matrix C

$$C = \begin{bmatrix} -1.742 & 0.390 & 0.118 & 2.347 & -2.043 & 1.614 & -0.316 & 0.125 & 0.037 & -0.141 \\ -0.304 & -0.088 & -0.159 & -0.141 & -0.419 & -0.070 & 0.064 & -0.355 & -0.106 & 0.132 \\ -0.207 & 0.189 & 0.002 & -2.854 & -0.934 & 0.275 & -0.256 & -0.822 & 0.159 & -0.186 \end{bmatrix}$$

Feedthrough matrix D

$$D = \mathbf{0}_{3\times 4}. \tag{27}$$

Even though Ko et al. (2003) does not state a sampling rate explicitly, considering other works from the same group such as Yeo et al. (2005), we can estimate the sampling rate to be 30 s

B.1 Data generation methodology

To simulate the system for controller development and testing, a data generation script was created. This script simulates the closed-loop response of the paper machine model for the grade change under PI control, introducing variability to create a rich dataset. The entire simulation, including the control law, operates on scaled data to normalize the variable ranges.

Scaling All variables are scaled to facilitate control and learning.

Controlled variables (CVs) – basis weight (Y_1), ash content (Y_2) and moisture content (Y_3) – are expressed in $\mathrm{g\,m^{-2}}$, %, and %, respectively. They are first converted to deviations from nominal values, $Y_{\mathrm{nom}} = [64.0\,\mathrm{g\,m^{-2}},\ 7.0\,\%,\ 9.0\,\%]^{\mathsf{T}}$, and then normalised:

$$y_{\mathrm{scaled}} = \frac{y_{\mathrm{phys}} - Y_{\mathrm{nom}}}{Y_{\mathrm{nom}}}, \tag{28}$$

$$y_{\mathrm{phys}} = Y_{\mathrm{nom}}\big(1 + y_{\mathrm{scaled}}\big). \tag{29}$$

Manipulated variables (MVs) are scaled to the range $[-1, 1]$ using the physical limits in Table 9.

Table 9: Physical limits used for MV scaling

MV	$u_{\min}$	$u_{\max}$
Stock flow ($\mathrm{L\,min^{-1}}$)	300	600
Filler flow ($\mathrm{L\,min^{-1}}$)	0	200
Steam pressure (bar)	1	3
Machine speed ($\mathrm{m\,min^{-1}}$)	800	1000

The forward and inverse maps are

$$u_{\mathrm{scaled}} = 2\,\frac{u_{\mathrm{phys}} - u_{\min}}{u_{\max} - u_{\min}} - 1, \tag{30}$$

$$u_{\mathrm{phys}} = u_{\min} + 0.5\,(u_{\max} - u_{\min})\big(u_{\mathrm{scaled}} + 1\big). \tag{31}$$

Simulation and control loop: The script executes 100 simulation runs, each for 250 time steps.

Initialization: Each run begins with a random initial state $x(0)$ drawn from a uniform distribution between -0.1 and 0.1.

PI controller with gain uncertainty: A PI controller is used to drive the system to the desired set-point. To ensure the data reflects realistic process variations, the controller gains (K_p, K_i) are randomized in each run. Base gains are multiplied by random factors drawn from specified uncertainty bands.

The control action u_{scaled} is calculated based on the scaled error $e_{\mathrm{scaled}} = Y_{\mathrm{sp,scaled}} - y_{\mathrm{scaled}}$:

$$u_{\mathrm{scaled}}(k) = u_{\mathrm{scaled,nom}} + K_p \cdot e_{\mathrm{loop}}(k) + \mathrm{int\_term}(k).$$

Here, e_{loop} is a four-element error vector (errors for Y_1, Y_2, Y_3, and Y_1 again) used to compute the four MV actions.

Anti-windup: An anti-windup mechanism is included. The integral term is only updated if the controller output is not saturated and the error is driving it further into saturation.

State update: The plant state is updated at each step. The control law computes the scaled MV u_{scaled}, which is then descaled to its physical value u_{phys}. The deviation from the nominal operating point, $\Delta u_{\mathrm{phys}} = u_{\mathrm{phys}} - U_{\mathrm{nom}}$, is used in the state-space equation

$$x(k+1) = A\,x(k) + B\,\Delta u_{\mathrm{phys}}(k).$$

The resulting state deviation is used to calculate the physical CV output, to which multiplicative noise is added before it is re-scaled for the controller.

Pairing. A diagonal four-loop PI structure is used with second basis-weight loop in a SIMO control arrangement, where stock flow U_1 and machine speed U_4 are driven by the basis-weight error e_{Y_1}; filler (clay) flow U_2 is driven by the ash-content error e_{Y_2}; and steam-header pressure U_3 is driven by the moisture error e_{Y_3}.

B.2 Reward function

Let $e_i(k)$ be the *scaled* tracking error of the i-th controlled variable considering the setpoints corresponding to the ongoing grade change scenario at step k, and define the instantaneous aggregate error

$$e(k) = \sum_{i=1}^{3} e_i^2(k).$$

Its one-step improvement is $\Delta e(k) = e(k-1) - e(k)$. For the action vector $u(k) = [u_1(k), u_2(k), u_3(k), u_4(k)]$ (scaled stock flow, filler flow, steam pressure and machine speed) and the step change $\Delta u(k) = u(k) - u(k-1)$, the reward is

$$r(k) = \alpha\Big[-e(k) + \exp\Big(-\tfrac{e(k)}{\sigma^2}\Big)\Big] + \Delta e(k) - \lambda \left\|\Delta u(k)\right\|_2^2, \tag{32}$$

with $\sigma = 0.1$. As in the previous case study, the first bracketed term rewards proximity to the set-point, the middle term rewards progress, and the last term discourages aggressive control moves.

B.3 Logged dataset

At each control interval, the simulator appends one row containing: identifiers (run, step); scaled CVs Y1, Y2, Y3 (basis weight, ash, moisture); scaled MVs U1−U4 (stock flow, filler flow, steam-header pressure, machine speed); error channels $e_1 - e_3$ and `err_metric` ($\sum_i e_i^2$); integrator states `int1`...`int4`; and the reward. All values are normalized (CVs, MVs) or dimensionless, so the CSV is ready for direct use by RL algorithms without further scaling.

B.4 Control scenario

The original study by Ko et al. (2003) identified seven distinct paper grades, labeled A through G, each defined by specific setpoints for the key quality variables: basis weight, ash content, and moisture content, as detailed in Table 10. For the control problem in this work, we focus on a single, challenging transition: a grade change from Grade A to Grade D. This scenario requires the controller to steer the process from a nominal state with a basis weight of 64 g/m^2 and an ash content of 7% to a new target state with a basis weight of 102 g/m^2 and an ash content of 17%. The moisture content setpoint remains constant at 9% for both grades. This represents a significant change in operating conditions, demanding a coordinated response from all four manipulated variables to manage the transition efficiently.

Table 10: Set point changes according to paper grades (Kao et al., 2025)

Paper grade	**A**	**B**	**C**	**D**	**E**	**F**	**G**
Basis weight (g/m^2)	64	49	78	102	63	100	78.5
Ash content (%)	7	7	17	17	9	17	14
Moisture content (%)	9	8	9	9	9	9	9