

Article

AES-RTO: An Adaptive Evolution Strategy for Real-Time Economic Optimization of High-Dimensional Chemical Processes

Bin Wang, Yifan Huang  and Qibing Jin *

Institute of Automation, Beijing University of Chemical Technology, Beijing 100020, China

* Correspondence: jinqb_buct@163.com

Abstract

Chemical processes operate under time-varying economic conditions, such as changes in raw-material prices and market demand. Such changes may shift the economically optimal operating point. For high-dimensional systems, real-time optimization becomes difficult because the solution time rises rapidly with the number of decision variables and can exceed the available control interval. To address that issue, this paper develops an adaptive evolution strategy-based real-time optimization framework, termed AES-RTO. The framework first applies economically weighted uniform manifold approximation and projection (UMAP) to project the high-dimensional operating space into a low-dimensional latent space while retaining the cost-related structure. AES searches this latent space by adjusting the step size, expanding the population when needed, and performing directionally guided long-range exploration. These operations broaden the search when the optimization stalls and help prevent early convergence in local regions. The suboptimal point obtained from the search is then supplied to a gradient-based RTO step as a warm start, reducing the iterations required for final refinement. The framework was evaluated on a single-stage continuous stirred-tank reactor (CSTR), a higher-dimensional cascade CSTR, and the Tennessee Eastman Process. Across these cases, it improved the initial search result relative to the derivative-free methods considered and remained applicable to processes with heterogeneous constraints. The results indicate that AES-RTO provides a computationally efficient route to economically favorable operating points on the benchmark models studied, under changing economic parameters.

Keywords: real-time optimization; adaptive evolution strategy; dimensionality reduction; time-varying economic optimization; manifold learning



Academic Editor: Massimo Caruso

Received: 26 August 2026

Revised: 13 October 2026

Accepted: 15 October 2026

Published: 18 October 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

In large-scale chemical plants, a moderate fluctuation in raw-material prices combined with a delayed response in the operating point can lead to a non-negligible daily economic loss, motivating the need for timely re-optimization. Production systems face time-varying economic objectives because resource prices, market demand, and operating constraints change over time. Therefore, timely adjustment of operating setpoints is important for maintaining competitiveness [1]. Alhazmi and Sarathy confirmed that chemical plants must continuously re-derive optimization policies from historical data to adapt to market and feedstock disturbances, rather than relying on a fixed steady-state target [2]. Economic model predictive control (EMPC) was developed to integrate time-varying objectives

directly into the predictive control layer, overcoming the decoupling lag of hierarchical RTO-MPC architectures [3,4], and has since matured into a standard framework with theoretical guarantees on feasibility and stability [5,6], validated in fluid catalytic cracking and packed-bed reactor systems [7,8].

However, most chemical production systems involve high-dimensional decision spaces with multiple manipulated variables. Biegler highlighted that multi-level optimization of large-scale nonlinear processes faces a fundamental conflict among model fidelity, dimensionality, and available computation time [9]. Wu et al. found that EMPC's computational burden grows rapidly with state dimension and horizon length [10,11], and Vaupel et al. showed that even ML-accelerated NMPC still exceeds minute-level real-time constraints for high-dimensional systems [12]. Krishnamoorthy proposed a sensitivity-based data-augmentation framework to ease this burden, but reported that its effectiveness still degrades as the number of manipulated variables increases [13].

Modifier-adaptation methods address plant-model mismatch through iterative measurement updates. However, their performance is still constrained by the dimensionality of the underlying steady-state optimization problem. Patrón and Ricardez-Sandoval proposed a directional modifier-selection scheme [14]; Bottari introduced a quadratic surrogate approximation [15]; and Zhang et al. developed a globally convergent trust-region framework [16]. All three, however, confirmed that modifier-adaptation remains fundamentally limited by the dimensionality of the underlying steady-state problem.

An alternative, lookup table-based RTO (LT-RTO), precomputes an offline library of operating points for sub-second retrieval [17,18]. Yang et al. demonstrated a trade-off between grid resolution and the curse of dimensionality: finer discretization expands the search space unmanageably as variables increase, while coarser grids introduce unacceptable error [19]. This difficulty arises partly because a uniform grid does not make use of the lower-dimensional structure that may exist in the optimal operating manifold.

Dimensionality reduction addresses this by exploiting a system's intrinsic low-dimensional structure. Healy and McInnes formalized uniform manifold approximation and projection (UMAP) as a manifold-learning technique preserving both local and global structure [20], later adopted for bioprocess soft-sensor development [21]. Hassanpour et al. showed that autoencoder embeddings can be used to calculate feasible MPC setpoints [22], while Marra et al. developed a latent-space NMPC formulation for partially observable systems [23]. Zhao et al. and Luo et al. also investigated reduced-order modeling and case-based dimensionality reduction for process control and optimization [24,25]. In vehicle powertrain modeling, physics-informed and theory-constrained data-driven models have been developed to keep the model itself interpretable [26,27]. Embeddings learned purely from data, such as UMAP, do not provide coordinates with physical meaning, so their value lies in the structure they preserve rather than in the interpretability of their coordinates. A remaining issue is that a promising latent-space point must be mapped back to the original space without losing feasibility or the relevant economic characteristics.

Derivative-free optimization provides an option when gradient information is unavailable or unreliable. SPSA lowers the number of function evaluations needed for gradient estimation, but the variance of this estimate may increase with dimension [28]. Bayesian optimization can improve global search through group-testing and coarse-to-fine strategies [29,30]. However, surrogate updates can remain costly for expensive high-dimensional problems. CMA-ES can lose covariance-learning efficiency in problems with low effective dimensionality, while particle-swarm approaches may require larger populations as the search dimension grows [31,32]. Hierarchical coarse-to-fine schemes are another way to limit the number of expensive evaluations. In powertrain co-optimization, for example, a population-based search coupled with a high-fidelity model provides pre-

liminary Pareto solutions, which are then refined with a surrogate model and Bayesian optimization [33,34].

To the best of our knowledge, dimensionality-reduction methods for RTO have seldom been combined with a latent-space search procedure that also recovers feasibility and supports online economic optimization within a single procedure. Such a procedure should work directly in a reduced space, maintain a reasonable balance between exploration and convergence, and return candidates that remain feasible and economically meaningful after inverse mapping.

This paper proposes an adaptive evolution strategy-based real-time optimization framework, referred to as AES-RTO, for moderately high-dimensional nonlinear processes. The framework combines manifold learning with adaptive evolutionary search. Economically weighted UMAP is used to compress the operating space into a low-dimensional latent space while retaining the cost-related structure. AES then explores this space using adaptive step sizing and directionally guided exploration. Its resulting candidate is subsequently used as a warm start for gradient-based RTO. The framework is tested on a single-stage continuous stirred-tank reactor (CSTR), a cascade CSTR, and the Tennessee Eastman Process.

The main contributions are the following: (1) We propose an RTO architecture coupling an economically weighted manifold embedding with population-based search in the reduced latent space, balancing search-space tractability and constraint feasibility via an inverse-mapping mechanism. (2) We design an adaptive evolution strategy that adjusts step size and population scale to cost-surface curvature and economic dynamics, achieving faster convergence without the population growth typically needed in high dimensions. (3) We demonstrate that AES-generated suboptimal points provide effective warm starts for downstream RTO refinement across CSTR systems and the Tennessee Eastman Process.

The rest of this paper is organized as follows: Section 2 presents the background; Section 3 describes the AES path-based RTO framework method, including process modeling, manifold-based workspace construction, cost surface generation, AES algorithm design, and RTO implementation; Section 4 validates the method on a single-stage CSTR, a higher-dimensional three-stage cascade CSTR with ablation and baseline comparisons, and the Tennessee Eastman Process as a cross-object generalization test; Section 5 concludes with the main findings.

2. Theoretical Framework of RTO-MPC and High-Dimensional Optimization Challenges

Real-time economic optimization of high-dimensional nonlinear processes is subject to severe computational and efficiency constraints. This section reviews three components relevant to the proposed framework, namely real-time optimization, dimensionality reduction, and path optimization in a low-dimensional space, and establishes a unified notational system for the AES-RTO framework presented in Section 3.

2.1. Real-Time Optimization

RTO aims to determine the steady-state operating point that minimizes economic cost under current process conditions, subject to process constraints. Its standard formulation is expressed as

$$\begin{aligned} & \min_{x,u} J(x,u;\theta) \\ \text{s.t. } & \begin{cases} f(x,u) = 0 \\ x \in X \\ u \in U \end{cases} \end{aligned} \quad (1)$$

where $J(\cdot)$ denotes the economic cost function, x denotes the process state variables, u denotes the manipulated variables, θ denotes the time-varying economic parameter vector associated with market and production conditions, $f(x, u) = 0$ denotes the equality constraint imposed by the process model, and X and U denote the feasible domains of the state and manipulated variables, respectively. Equation (1) is commonly solved using gradient-based nonlinear programming (NLP) methods such as sequential quadratic programming (SQP). Such methods converge reliably when the initial point lies close to the optimum, but their convergence rate decreases substantially as the decision-variable dimension increases or as the initial point deviates further from the optimum. In high-dimensional problems, the time required for a cold-start optimization may exceed the computational window allotted to a single RTO cycle. Providing a high-quality warm-start point to the NLP solver is therefore an important consideration in the engineering implementation of high-dimensional RTO.

2.2. Dimensionality Reduction Methods

High-dimensional decision variables substantially increase the online computational burden of RTO. Dimensionality reduction constructs a nonlinear mapping from the original operating space to a low-dimensional latent space, compressing the search dimension while retaining relevant geometric and economic information. Its general form is expressed as

$$z = \varphi(u), \quad z \in Z \subset \mathbb{R}^d, \quad d \ll m \quad (2)$$

where z denotes the latent-space variable, $\varphi(\cdot)$ denotes the nonlinear mapping from the m -dimensional operating space to the d -dimensional latent space, and Z denotes the feasible region in the latent space. This transformation does not replace the original optimization problem; rather, it confines the coarse search to a region of substantially reduced volume, allowing subsequent optimization to proceed at lower computational cost. An effective low-dimensional representation retains the principal economic structure of the original problem, so that a favorable candidate point identified in Z remains economically meaningful once mapped back to the original space.

2.3. Path Optimization in the Low-Dimensional Space

Identifying a favorable operating point in the low-dimensional latent space requires a balance between global exploration and local convergence speed. Because gradient information is not always available or reliable in this setting, a population-based, derivative-free search strategy is adopted. Its general iterative form is expressed as

$$z_k^{(i)} = z_k + \sigma_k \cdot d_k^{(i)}, \quad i = 1, \dots, \lambda \quad (3)$$

where z_k denotes the population center at generation k , σ_k denotes the search step size at generation k , and $d_k^{(i)}$ denotes the random search direction of the i -th offspring. In each generation, candidate solutions are evaluated by their fitness, defined as the economic cost obtained after mapping back to the original space, and the individual achieving the lowest cost is retained as the center of the next generation. This iterative process guides the search toward low-cost regions of the latent space. The resulting candidate point then serves as the initial guess for the RTO layer. Because this point is closer to a favorable region, the constrained optimizer can focus on local refinement rather than a cold-start global adjustment. Section 3 presents the complete design of the proposed AES-RTO method on this basis.

3. AES-Path-Based RTO Method

This section presents the proposed AES-RTO method in four parts: nonlinear projection of the high-dimensional operating space, construction of a low-dimensional economic landscape, design of the Adaptive Evolution Search algorithm, and the two-layer RTO-MPC closed-loop architecture.

3.1. Nonlinear Projection of the High-Dimensional Space

For a general operating-point optimization problem, let m denote the dimension of the manipulated-variable vector u , subject to the feasible-domain constraint $u \in U$. Under time-varying economic parameters θ , performing RTO directly in the high-dimensional original space is subject to the curse of dimensionality, whereby the computational cost grows rapidly with dimension. To address this limitation, UMAP is used to project the m -dimensional operating space onto a low-dimensional embedded space Z , where the dimension of Z , denoted d , satisfies $d \ll m$, establishing the nonlinear mapping

$$z = \varphi_{\text{UMAP}}(u), \quad \varphi_{\text{UMAP}}: U \rightarrow Z \quad (4)$$

3.1.1. Training Data Sampling and Dimensionality Reduction Strategy

The quality of the dimensionality-reduction mapping depends on the extent to which the training samples cover the feasible operating space and on their economic relevance. In contrast to approaches that construct the embedding solely from optimal solutions obtained under a specific set of economic parameters, a two-stage sampling strategy is adopted here. The method first performs uniform random sampling across the operating space. This step provides coverage of the feasible domain and reduces the risk that the mapping overfits one economic scenario. The method then applies short local optimization to a subset of samples. Therefore, the training set retains broad geometric coverage while including more economically relevant points. The uniform-sampling stage is expressed as

$$u^{(i)} = u_{\min} + \text{rand}(0,1) \odot (u_{\max} - u_{\min}), \quad i = 1, \dots, N \quad (5)$$

where $u_{\min}$ and $u_{\max}$ denote the lower and upper bounds of each manipulated variable, respectively, $\text{rand}(0,1)$ denotes a uniformly distributed random number on the interval $[0,1]$, and N denotes the total number of training samples. To improve the economic relevance of the sampled points, a short local optimization under economic parameters randomly perturbed around the reference values is applied to a subset of the samples:

$$u_{\text{ref}}^{(j)} = \mathcal{L} \left(u^{(j)}; \theta_{\text{nom}}, K_{\max} \right), \quad j \in S \quad (6)$$

where $\mathcal{L}(\cdot)$ denotes the truncated local-optimization operator, θ_{nom} denotes the reference economic parameter vector, $K_{\max}$ denotes the maximum number of local-optimization iterations, and S denotes the subset of samples selected for refinement. This semi-optimized sampling strategy preserves overall coverage of the feasible domain while directing part of the samples toward regions of favorable economic performance, so that the training data possess both geometric diversity and economic sensitivity.

To ensure numerical stability, a hybrid normalization scheme is applied prior to dimensionality reduction. Variables of higher economic sensitivity retain their original physical magnitude or are assigned a larger weight; the remaining manipulated variables are linearly normalized according to their feasible range:

$$\tilde{u}_i = \frac{u_i - u_i^{\min}}{u_i^{\max} - u_i^{\min}} \quad \text{or} \quad \tilde{u}_i = w_i \cdot u_i \quad (7)$$

This treatment allows UMAP to weight the neighborhood graph according to the true physical magnitude of the economically sensitive dimensions, avoiding the possibility that other manipulated variables dominate the neighborhood structure owing to differences in their units. The weighting scheme in Equation (7) is determined directly from the relative magnitudes of the economic weight coefficients θ in the cost function: variables associated with larger coefficients in θ retain their original physical scale, while the remaining variables are linearly normalized. Compared with linear dimensionality-reduction methods such as principal component analysis (PCA), UMAP more effectively preserves the nonlinear neighborhood relationships present in the high-dimensional space while retaining the global organization required for latent-space search. A quantitative comparison between UMAP and PCA, in terms of both reconstruction quality and downstream search performance for the single-stage CSTR case, is provided in Appendix A (Table A1, Figure A1).

3.1.2. Latent-Space Feasibility Constraints

The embedded samples obtained after dimensionality reduction define an empirical feasible region in the latent space. To ensure that subsequent search points in the low-dimensional space can be reliably mapped back to feasible operating points, the latent-space boundary is determined from the extrema of each embedding coordinate, with a moderate expansion applied:

$$Z = [\min(z) - \eta \cdot \text{range}(z), \max(z) + \eta \cdot \text{range}(z)] \quad (8)$$

where the margin coefficient is set to 10%. This moderate boundary expansion allows limited extrapolation near the boundary of the training manifold, while avoiding excessive extrapolation into latent-space regions with sparse representation, where the inverse mapping would become unreliable.

3.1.3. Inverse Mapping and Feasibility Guarantee

A central difficulty of manifold-based dimensionality reduction concerns the reliable mapping of a search result z from the low-dimensional embedded space back to the m -dimensional original operating space while satisfying process constraints. A weighted k -nearest neighbor (KNN) inverse mapping scheme is adopted, in which the operating point is reconstructed from a weighted combination of multiple feasible samples in the neighborhood of the query point, rather than from a single nearest sample, thereby improving the smoothness and robustness of the reconstruction:

$$\hat{u} = \sum_{k \in N(z)} w_k \cdot u^{(k)} \quad (9)$$

where $N(z)$ denotes the index set of the KNN of the query point z in the latent-space training set, and the weights are assigned according to inverse-distance confidence:

$$w_k = \frac{1/d_k}{\sum_{k' \in N(z)} 1/d_{k'}} \quad (10)$$

The distances in Equation (10) are Euclidean distances in the latent space, and 10^{-8} is added to each distance to avoid division by zero. To further improve the reliability of the reconstruction, a two-stage filtering mechanism is introduced. Feasible neighboring samples satisfying the feasibility condition are prioritized in the weighted average; if the number of feasible neighboring points is insufficient, or their confidence is too low, the scheme reverts to a weighted average over all k neighboring samples. The reconstructed

result is finally projected onto the manipulated-variable domain to ensure satisfaction of the box constraints:

$$u^* = \text{Proj}_U(\hat{u}) \quad (11)$$

The projection $\text{Proj}_U(\cdot)$ in Equation (11) guarantees satisfaction of the box constraints on the manipulated variables, but it does not analytically guarantee feasibility with respect to nonlinear process constraints. Therefore, the final reconstructed candidate is evaluated using a case-specific feasibility screening procedure before downstream RTO refinement. If the candidate does not satisfy the applicable feasibility criteria, it is replaced by the nearest simulation-verified feasible training sample among its latent-space neighbors. The feasibility diagnostics for the case studies are summarized in Section 4 and Appendix B.

The inverse-mapping strategy combines manifold-space search with a feasibility-oriented reconstruction process. It keeps the computational cost of latent-space exploration low while ensuring that the mapped operating point is feasible in the original space, thereby providing a favorable initial condition for the refinement step at the RTO layer. The reconstruction accuracy of this scheme was quantitatively assessed via leave-one-out validation on the training set; the results are reported in Appendix A (Table A1). Feasibility with respect to the nonlinear constraints instead relies on the fact that the training samples used for the weighted-KNN reconstruction are themselves feasible, simulation-verified operating points.

3.2. Construction of the Low-Dimensional Economic Landscape

After dimensionality reduction, the originally high-dimensional, non-convex economic optimization problem is transformed into a search problem on the low-dimensional manifold Z . To support visualization and the design of the search strategy, a $(d + 1)$ -dimensional economic landscape is constructed on the latent space, in which the first d dimensions correspond to the embedding coordinates and the $(d + 1)$ -th dimension represents the economic cost.

3.2.1. Grid Strategy for Landscape Generation

Within the feasible domain of the latent space, a regular d -dimensional grid is constructed centered at the current search starting point. For visualization purposes, $d = 2$ is used as an illustrative example. Let ξ_0 denote the starting point, r the search radius along each dimension, and n the number of points sampled uniformly along each dimension; the coordinates of each grid node are then given by

$$\xi_{i,j} = \xi_0 + (i \cdot \Delta_1, j \cdot \Delta_2), i, j = 0, \dots, n - 1 \quad (12)$$

For $d = 2$, this generates a $n \times n$ grid; for $d > 2$, the same construction extends naturally to a d -dimensional hypercube grid.

3.2.2. Landscape Height Calculation

For each grid node z , the landscape height is obtained by mapping z back to the original operating space through the inverse mapping and evaluating the current economic objective function:

$$h(\xi) = J(\text{Proj}_U(\Psi(\xi)); \theta) \quad (13)$$

$$J = \sum_l w_l \cdot c_l(u) \quad (14)$$

where $\Psi(\cdot)$ denotes the inverse mapping function, c_l denotes the l -th economic performance component, such as energy consumption, production rate, or safety margin, and w_l denotes the corresponding economic weight. This formulation preserves the generality of the

landscape definition, accommodating objective functions of different concrete forms across case studies.

3.2.3. Landscape Smoothing and Anomaly Handling

The computed landscape surface may contain numerical anomalies, such as extreme values or discontinuities arising from uncertainty in the inverse mapping, which distort the visualization scale and interfere with interpretation of the landscape structure. A cost upper-bound threshold $h_{\max}$ is therefore defined at the rendering stage:

$$c_{lo} = P_2(h), \quad c_{hi} = P_{98}(h) \quad (15)$$

where $P_2(h)$ and $P_{98}(h)$ denote the 2nd and 98th percentiles of the computed landscape heights, respectively. Grid nodes are rendered using the color range $[c_{lo}, c_{hi}]$, which is used for visualization only.

3.2.4. Economic Interpretation of the Landscape

The constructed $(d + 1)$ -dimensional landscape has a clear geometric and economic interpretation, summarized in Table 1. The gradient of the landscape points in the direction of the steepest cost increase, expressed as

$$\nabla h(\xi) = \frac{\partial h}{\partial \xi} \quad (16)$$

Table 1. Economic interpretation of landscape features.

Landscape Feature	Mathematical Description	Economic Meaning
Deep valley	h value minimum	Global or local optimum
Plateau	h value constant	Constant economic benefit within the region
Steep slope	$ \nabla h $ large	Sensitive parameter region
Multiple peaks	Multiple isolated minima	Multiple local optima exist

Equation (16) describes the local geometry of the landscape, and the AES search itself is gradient-free. This formulation clarifies why certain regions of the latent space are readily traversed, whereas others are more likely to trap the search in a local optimum or produce sharp fluctuations in cost.

3.3. AES Optimization Algorithm

3.3.1. Basic Evolution Loop of AES

Evolution strategies are population-based, gradient-free optimization methods that are well suited to multi-modal, non-convex, and non-smooth objective functions. AES combines adaptive step sizing, population expansion, long-range exploration, and annealing-style acceptance with coupled triggering conditions specifically tailored to search in an economically weighted latent space, and uses its output as a warm start for downstream gradient-based refinement rather than as a standalone final optimizer, which shifts its design objective from approaching the global optimum to rapidly identifying a favorable feasible neighborhood.

On the low-dimensional economic landscape constructed in Section 3.2, the AES algorithm performs its search through an iterative cycle of population generation, evaluation, and selection. Because the optimizer has no prior knowledge of the overall landscape topology, this process resembles a blind individual descending a mountain: At each step, the local slope can be sensed, but the location of the globally lowest valley cannot be observed in advance, as illustrated in Figure 1. Unlike swarm-intelligence algorithms

that are initialized randomly across the entire feasible domain, the first generation of the AES population is instead sampled locally around the operating point that preceded the economic parameter change.



Figure 1. A blind person is looking for the way down the mountain.

The evolution process at generation k proceeds as follows. First, λ offspring are sampled around the current optimal point z_k^{best} :

$$z_k^{(i)} = z_k^{\text{best}} + \sigma_k \cdot \varepsilon^{(i)}, \quad \varepsilon^{(i)} \sim \mathcal{N}(0, I_d), \quad i = 1, \dots, \lambda \quad (17)$$

where σ_k denotes the adaptive step size at generation k , and $\varepsilon^{(i)}$ denotes a d -dimensional standard Gaussian random vector. Offspring exceeding the latent-space boundary are then clipped to the feasible-domain boundary:

$$z_k^{(i)} = \text{clip} \left(z_k^{(i)}, Z_{\min}, Z_{\max} \right) \quad (18)$$

The fitness of each offspring is evaluated through the inverse mapping and the corresponding economic cost:

$$J_k^{(i)} = h \left(z_k^{(i)} \right) \quad (19)$$

Among the λ offspring, the individual with the lowest cost is selected as the center of the next generation:

$$z_{k+1} = \arg \min_i J_k^{(i)}, \quad i = 1, \dots, \lambda \quad (20)$$

and the historical optimum is updated whenever a strict improvement is obtained:

$$z_{k+1}^{\text{best}} = \begin{cases} z_{k+1}, & J(z_{k+1}) < J(z_k^{\text{best}}), \\ z_k^{\text{best}}, & \text{otherwise.} \end{cases} \quad (21)$$

This basic loop forms the search backbone of AES. The adaptive mechanisms introduced in Section 3.3.2 further enhance the exploration capability of the algorithm to avoid premature convergence to a shallow local optimum.

3.3.2. Mechanisms for Escaping Local Optima

Economic landscapes typically contain multiple local minima, and a purely greedy evolution process is prone to premature convergence to the basin of attraction nearest the starting point. AES addresses this limitation through three coordinated mechanisms, namely adaptive population expansion, an adaptive initial step size, and directionally guided long-range exploration (DGLE), together with an annealing-style acceptance criterion.

Adaptive Population Expansion

When the search stagnates, AES expands the offspring population to increase directional diversity and local coverage. The expansion is controlled by a preset schedule so that AES can escape a local basin without maintaining an excessively large population throughout the search:

$$\lambda_k = \lambda_0 + (\lambda_{\max} - \lambda_0) \cdot g\left(\frac{s_k}{T}\right) \quad (22)$$

where λ_0 denotes the baseline population size, $\lambda_{\max}$ denotes the maximum population size, T denotes the total number of iterations, and s_k denotes the number of consecutive generations without significant improvement. When s_k is below a set threshold, the algorithm proceeds at its normal optimization pace; when the threshold is exceeded, the algorithm enters an escape mode and the population expands according to a preset schedule; if stagnation persists beyond a further, longer limit, the current optimization phase is terminated and the historical optimum is retained.

Adaptive Initial Step Size

The search step size serves different roles at different stages of optimization: A larger step size is used in the early stage to traverse the main features of the landscape for coarse-grained search; a moderate step size is used in the middle stage to balance exploration and exploitation while maintaining convergence speed; and a smaller step size is used in the later stage to enable fine-grained search and ensure convergence precision. The adaptive initial step size is accordingly defined as

$$\sigma_k = \sigma_0 \cdot \psi\left(\frac{k}{T}\right) \quad (23)$$

where $\psi(\cdot)$ denotes a step-size adjustment function that decreases monotonically with iteration progress.

Directionally Guided Long-Range Exploration

In the absence of directional guidance, a purely greedy search proceeding along the downhill direction is prone to settling in a shallow depression among the surrounding peaks, i.e., becoming trapped in a local optimum. To address this limitation, after the direction of population variation is determined, AES evaluates not only the cost value after the step size σ_k but also the terrain farther along the same direction, with the weight assigned to this farther evaluation decaying as the iteration count increases. This mechanism is expressed as

$$d_k^{(i)} = (1 - \beta_k) \cdot d_{\text{local}}^{(i)} + \beta_k \cdot d_{\text{long}}^{(i)} \quad (24)$$

Candidate points identified through this evaluation may be accepted for their exploratory value even without an immediate improvement in cost, through an annealing-style acceptance criterion:

$$P(\text{accept}) = \exp\left(-\frac{\Delta J}{T_k}\right) \quad (25)$$

where T_k denotes an exploration temperature that decays with iteration. The concrete parameter values used in the experiments, the explicit functional forms of $\psi(\cdot)$, β_k , and T_k , and a one-factor-at-a-time sensitivity analysis for the parameters λ_0 , λ_{max} , the stagnation threshold, T_{init} , and τ , are provided in Appendix A (Tables A2 and A3). In brief, AES-RTO's performance is broadly robust to these parameters within the tested ranges, with a larger maximum population size improving both mean solution quality and run-to-run consistency, and the exploration temperature exhibiting a saturation effect above a moderate threshold.

The DGLE mechanism drives the search to probe neighboring basins beyond the current shallow valley, while the annealing-style acceptance criterion prevents the algorithm from behaving overly conservatively when encountering a shallow saddle or a temporary increase in cost. During stagnation, the expanded population generates candidates around the current point, while DGLE probes promising directions beyond the local basin. The annealing-style acceptance rule allows limited exploratory moves even when they do not give an immediate cost decrease. Together, these mechanisms improve the chance of escaping a shallow local optimum without making the search overly conservative.

3.4. Two-Layer RTO Architecture and MPC Closed-Loop Execution

3.4.1. Real-Time Generation of Suboptimal Points

When raw-material prices or market demand change, the economic parameters θ change accordingly, and the previously optimal operating point is no longer optimal under the new conditions. Computing the strictly optimal operating point u^* often requires considerable time, which increases further with system complexity; continued tracking of the previous target during this computation results in accumulated economic loss. To mitigate this limitation, the AES path search presented in Sections 3.1–3.3 is used to provide a feasible operating point $\hat{u}$ with already improved economic performance before the RTO layer completes its computation of u^* , thereby reducing economic loss during the waiting period.

3.4.2. Rigorous Constraint Optimization at the RTO Layer

In parallel with the AES search, a second stage of full optimization solves the standard constrained nonlinear programming problem:

$$\begin{aligned} & \min_u J(u; \theta) \\ & \text{s.t.} \begin{cases} f(x, u) = 0 \\ u \in U \\ x \in X \end{cases} \end{aligned} \quad (26)$$

The Sequential Quadratic Programming algorithm implemented in the MATLAB `fmincon` solver is used to solve the RTO-layer problem, with the parameter settings listed in Table 2. All experiments were conducted on a computer equipped with an AMD Ryzen 5 3500X processor (Advanced Micro Devices, Inc., Santa Clara, CA, USA) and 32 GB of memory (Maxsun Technology Co., Ltd., Guangzhou, China), running the 64-bit Windows 10 Professional operating system (Microsoft Corporation, Redmond, WA, USA) and MATLAB R2020b (The MathWorks, Inc., Natick, MA, USA).

Table 2. RTO-layer solver settings.

Parameter	Value	Definition
MaxIterations	500	Maximum of 500 iterations
MaxFunctionEvaluations	2000	Maximum of 2000 function evaluations
TolFun	10^{-10}	Cost-function precision
TolX	10^{-10}	Operating-point space precision
TolCon	10^{-9}	Constraint-satisfaction precision

A cold start of the RTO layer from the original operating point u_0 generally requires more iterations to converge, increasing computation time. Initializing the RTO layer instead from the suboptimal point $\hat{u}$ obtained from the AES search accelerates convergence, because $\hat{u}$ already lies within the neighborhood of the optimum and fmincon performs only local refinement. Because $\hat{u}$ and u^* are both obtained under the same cost function, they typically do not differ substantially, so the switching of the target operating point delivered to the MPC layer remains relatively smooth.

4. Experiments and Result Analysis

The proposed AES-RTO framework was evaluated on three case studies of increasing complexity. A single-stage CSTR of moderate dimension was first used to compare AES against other latent-space search algorithms, facilitating visualization of the optimization trajectory and convergence behavior. A three-stage cascade CSTR was then introduced as a higher-dimensional case study, in which inter-stage concentration coupling and increased dimensionality jointly raise the difficulty of direct search; this system was used for ablation experiments and an overall baseline comparison. Finally, the Tennessee Eastman Process (TEP) was selected as a cross-object validation case, to assess whether the framework generalizes beyond CSTR-type systems to a classical industrial process with heterogeneous constraints and nonlinear steady-state characteristics.

To quantify the feasibility properties of the UMAP-KNN inverse mapping, three query sets were evaluated for each case study: AES offspring candidates generated during latent-space search, uniformly sampled points within the expanded latent domain, and boundary-focused latent points. For the CSTR cases, feasibility was assessed by independently solving the steady-state process equations without post-solution state clipping. For the 6D CSTR, all 6500 reconstructed points were directly process feasible. For the 16D cascade CSTR, 98.30–99.80% of reconstructed points were directly feasible after bound projection; the remaining 0.20–1.70% slightly exceeded the equality-residual tolerance but had zero inequality-constraint violation. Local feasible-neighbor repair yielded a final process-feasible rate of 100% for all query sets. For the TEP case, all 6500 reconstructed points satisfied the hard constraints evaluated by the local surrogate model. Detailed diagnostics are reported in Appendix B.

4.1. Single-Stage CSTR

4.1.1. Model Description

A non-isothermal CSTR in which the liquid-phase first-order reaction $A \rightarrow B$ occurs was used as the first benchmark object. The reactor is surrounded by a jacket through which a circulating cooling medium regulates the system temperature. The state vector is defined as $x = [C_A, T]^T$, where C_A denotes the reactant concentration and T denotes the reactor temperature. Six manipulated variables were considered: feed flow rate F , jacket

cooling temperature T_c , feed concentration C_0 , effective reaction volume V , stirring speed N , and system pressure P , giving the control vector

$$\mathbf{u} = [F, T_c, C_0, V, N, P]^T \in \mathbb{R}^6 \quad (27)$$

The reactor configuration, including the cooling jacket and circulating medium, is shown in Figure 2. The physical parameters used in the reactor model are summarized in Table 3. Under the assumptions of ideal mixing and constant physical parameters, the reactor is governed by the following mass- and energy-balance equations:

$$\begin{aligned} \frac{dC_A}{dt} &= \frac{F}{V}(C_0 - C_A) - k(T)C_A \\ \frac{dT}{dt} &= \frac{F}{V}(T_0 - T) + Q_{\text{react}} + Q_{\text{cool}} \end{aligned} \quad (28)$$

where the reaction-rate constant follows the Arrhenius relation, and the heat-generation and cooling terms are, respectively,

$$\begin{aligned} k(T) &= k_0 \exp\left(-\frac{E_a}{R_g T}\right) \\ Q_{\text{react}} &= \frac{-\Delta H}{\rho C_p} k(T) C_A \\ Q_{\text{cool}} &= \frac{UA}{\rho C_p V} (T_c - T) \end{aligned} \quad (29)$$

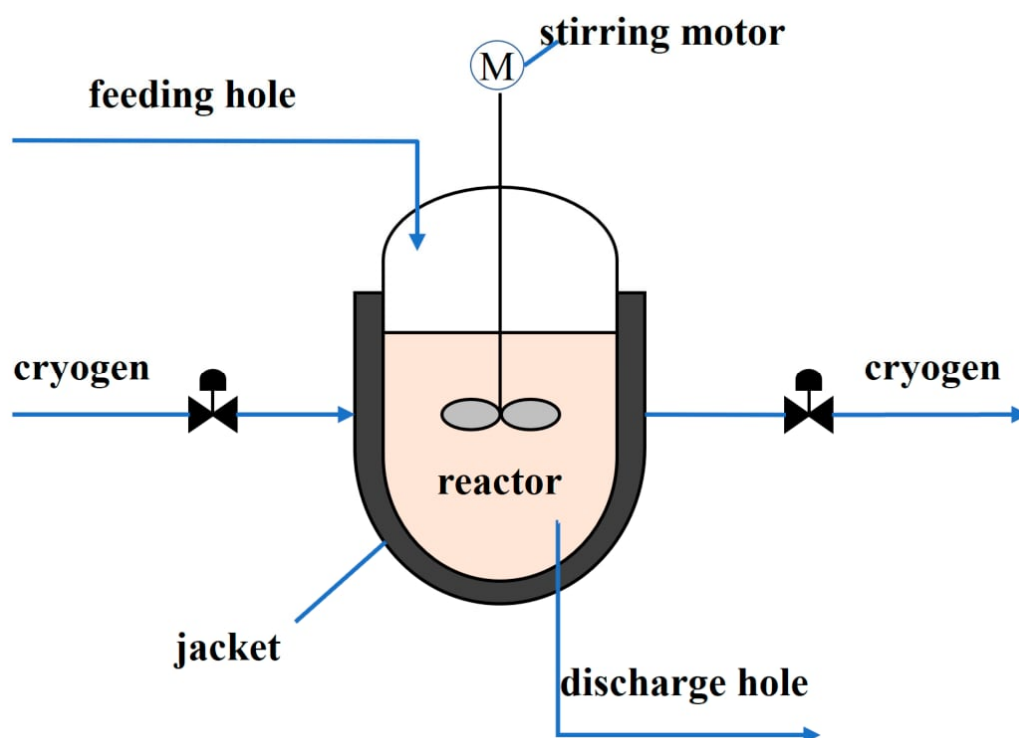


Figure 2. Single-stage non-isothermal CSTR structure diagram.

The six manipulated variables satisfy box constraints of the form $\mathbf{u}_{\min} = [0.5, 250, 0.5, 50, 100, 1.0]^T$ and $\mathbf{u}_{\max} = [8.0, 350, 3.0, 200, 500, 5.0]^T$, with units of m^3/h , K, kmol/m^3 , m^3 , rpm, and atm. The steady-state economic objective function accounts for cooling energy consumption, deviation from the target conversion, temperature deviation, volume utilization, stirring cost, and pressure cost:

$$\begin{aligned} J_E(\mathbf{u}) &= \theta_1 T_c^2 + \theta_2 (\text{conv} - 0.05)^2 + \theta_3 (T^{\text{ss}} - 360)^2 \\ &\quad + \frac{\theta_4}{100} (V - 125)^2 + \frac{\theta_5}{1000} N^2 + 0.1\theta_6 (P - 2.5)^2 \end{aligned} \quad (30)$$

where $\text{conv} = 1 - C_A^{ss}/C_0$ denotes the steady-state conversion, and C_A^{ss} denotes the steady-state reactant concentration. An additional soft safety-penalty term is applied when T exceeds the safe temperature bound. The economic weight vector $\theta = [\theta_1, \theta_2, \theta_3]^\top$ takes the nominal value $\theta^{\text{nom}} = [10^{-7}, 5 \times 10^{-8}, 10^{-6}]^\top$ under baseline conditions and the perturbed value $\theta^{\text{new}} = [5 \times 10^{-8}, 3 \times 10^{-8}, 8 \times 10^{-7}]^\top$ to represent a simultaneous change in energy price and product value.

Table 3. Physical parameters of the single-stage CSTR.

Parameter	Symbol	Value	Unit
Pre-exponential factor	k_0	3×10^7	h^{-1}
Activation energy	E_a	5×10^4	$\text{kJ} \cdot \text{kmol}^{-1}$
Gas constant	R_g	8.314	$\text{kJ} \cdot \text{kmol}^{-1} \cdot \text{K}^{-1}$
Heat of reaction	ΔH	-5×10^4	$\text{J} \cdot \text{kmol}^{-1}$
Liquid density	ρ	1000	$\text{kg} \cdot \text{m}^{-3}$
Specific heat	C_p	4184	$\text{J} \cdot \text{kg}^{-1} \cdot \text{K}^{-1}$
Overall heat-transfer coeff. area	UA	1.5×10^6	$\text{J} \cdot \text{h}^{-1} \cdot \text{K}^{-1}$
Feed temperature	T_0	300	K

4.1.2. Latent-Space Search

After the economic parameters change from θ^{nom} to θ^{new} , the cost of the current operating point changes from $J_{\text{old}} = 0.011109$ to $J_{\text{new}} = 0.006026$ under the new conditions, indicating that re-optimization is required. A two-dimensional latent space is used for this case. Five derivative-free algorithms, namely DESGD [35], SPSA [28], ES [36], Bandit [37], and AES, were embedded in the same two-dimensional UMAP latent space and initialized from the same starting point, with an identical function-evaluation budget.

Table 4 summarizes the first-layer results, while Figures 3 and 4 show the search trajectory and convergence behavior. The purple line indicates the AES search trajectory. The solid lines of Figure 4 depict the mean convergence curves, and the shaded bands represent the convergence range across the 50 repeated runs. AES achieved the largest improvement among the five methods. The trajectory includes a long-range move generated by DGLE, which enables the search to leave the initial basin. After this move, the adaptive step-size rule narrows the search region, and population expansion supports subsequent local refinement.

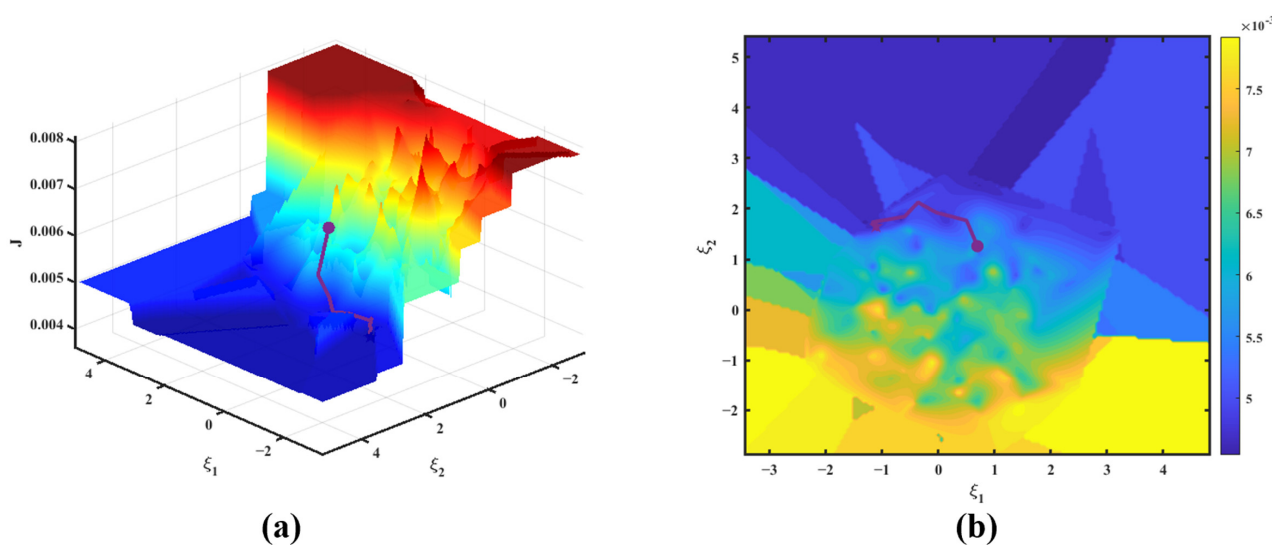
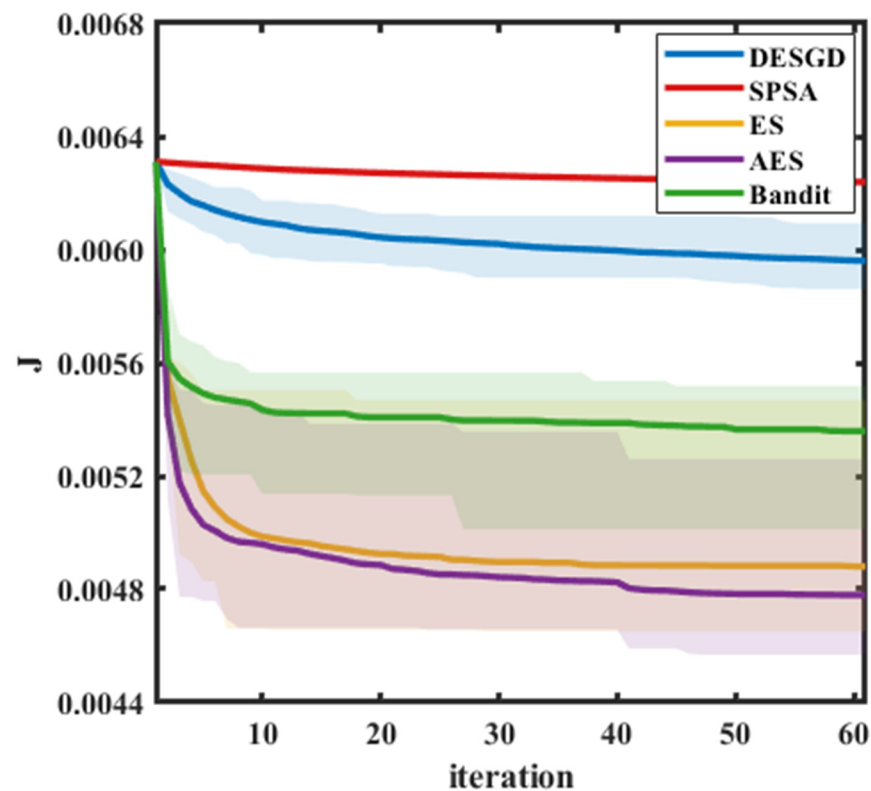


Figure 3. Optimization trajectory of AES in UMAP latent space (a) and visualization cost surface (b).

Table 4. First-layer latent-space search results (50 runs).

Algorithm	Final Cost J (10^{-4})	Improvement (%)	Mean Time (s)
DESGD	59.43 ± 0.62	1.38 ± 1.03	0.05
SPSA	62.44 ± 0.05	-3.62 ± 0.08	0.10
ES	48.53 ± 0.62	19.47 ± 1.03	0.24
AES	46.69 ± 0.32	22.52 ± 0.53	0.04
Bandit	53.77 ± 0.29	10.77 ± 0.48	0.26

**Figure 4.** Comparison of convergence of five subspace search algorithms on single CSTR.

4.2. Three-Stage Cascade CSTR

4.2.1. Model Description

To evaluate the framework under higher dimensionality and structural coupling, three CSTRs of the type described in Section 4.1 were connected in series, such that the outlet concentration of stage $r + 1$ serves as the inlet concentration of stage $r + 1$. The first stage has six independently adjustable variables, while the second and third stages each have five, as their inlet concentration is determined by the steady state of the preceding stage. The resulting 16-dimensional control vector is

$$\begin{aligned}
 \mathbf{u}_{16} &= [\mathbf{u}_1^\top, \mathbf{u}_2^\top, \mathbf{u}_3^\top]^\top \in \mathbb{R}^{16}, \mathbf{u}_1 \in \mathbb{R}^6, \mathbf{u}_2, \mathbf{u}_3 \in \mathbb{R}^5 \\
 \mathbf{u}_1 &= [F_1, T_{c,1}, C_{0,1}, V_1, N_1, P_1]^\top \\
 \mathbf{u}_2 &= [F_2, T_{c,2}, V_2, N_2, P_2]^\top \\
 \mathbf{u}_3 &= [F_3, T_{c,3}, V_3, N_3, P_3]^\top
 \end{aligned} \tag{31}$$

Given the higher dimensionality of this case and the absence of a corresponding visualization requirement, a three-dimensional latent space is used here, retaining more geometric and economic information than the two-dimensional embedding used in the single-stage CSTR case. The cascade configuration of the three reactor stages is illustrated in Figure 5. Each stage follows the same mass- and energy-balance relationships as the single-

stage CSTR, differing only in stage-specific parameter values. The inter-stage coupling constraint is

$$C_{0,2} = C_{A,1}^{ss}, \quad C_{0,3} = C_{A,2}^{ss} \quad (32)$$

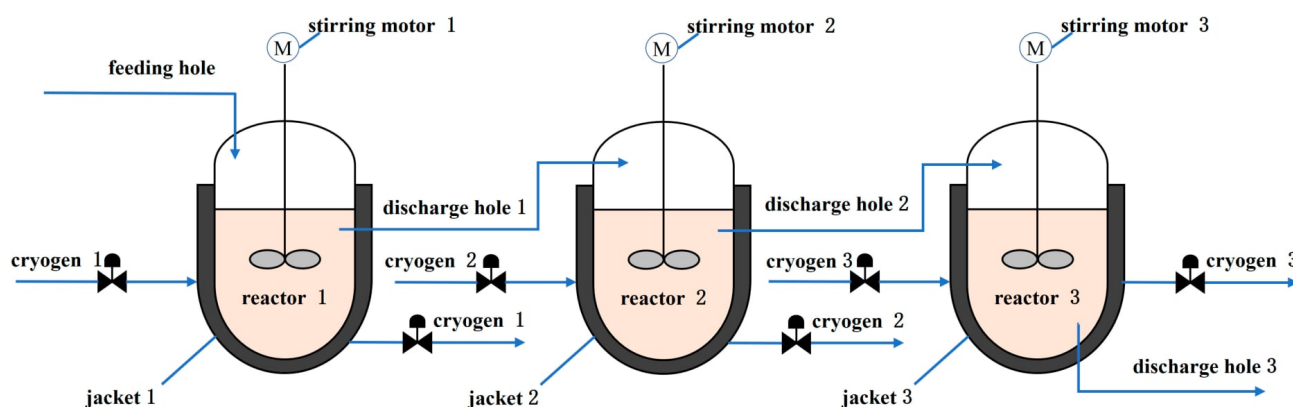


Figure 5. Three-stage cascade CSTR structure diagram.

Each stage has an economic cost function of the same structural form as Equation (30), with an independent weight vector $\theta_r = [\theta_{r,1}, \theta_{r,2}, \theta_{r,3}]^T$, and an additional safety penalty when the stage temperature exceeds its safe bound. The total system cost aggregates the cost of the three stages together with a boundary penalty term:

$$J_r(\mathbf{u}_r, \mathbf{x}_r^{ss}) = \theta_{r,1} T_{c,r}^2 + \theta_{r,2} (\text{conv}_r - 0.05)^2 + \theta_{r,3} (T_r^{ss} - 360)^2 + \frac{\theta_{r,1}}{100} (V_r - 125)^2 + \frac{\theta_{r,1}}{1000} N_r^2 + 0.1 \theta_{r,3} (P_r - 2.5)^2 \quad (33)$$

An additional safety penalty of $10^4 \max(0, T_r^{ss} - 400)^2$ is imposed for the r -th stage when $T_r^{ss} > 400$ K. The overall system economic cost aggregates the costs of the three stages together with the boundary penalty term.

$$J_E(\mathbf{u}_{16}) = \sum_{r=1}^3 J_r + J_{\text{penalty}} \quad (34)$$

$$J_{\text{penalty}} = 10^4 \sum_{i=1}^{16} \left[\max(0, u_{i,\min} - u_i)^2 + \max(0, u_i - u_{i,\max})^2 \right]$$

The nominal and disturbed economic weighting parameters for each stage are listed in Table 5.

Table 5. Nominal and perturbed economic weights for each stage of the cascade CSTR.

Stage	$\theta_{r,1}^{\text{nom}}$	$\theta_{r,2}^{\text{nom}}$	$\theta_{r,3}^{\text{nom}}$	$\theta_{r,1}^{\text{new}}$	$\theta_{r,2}^{\text{new}}$	$\theta_{r,3}^{\text{new}}$
R1	1.0×10^{-7}	5.0×10^{-8}	1.0×10^{-6}	1.2×10^{-7}	4.0×10^{-8}	1.1×10^{-6}
R2	1.0×10^{-7}	5.0×10^{-8}	1.0×10^{-6}	0.9×10^{-7}	6.0×10^{-8}	0.9×10^{-6}
R3	1.0×10^{-7}	5.0×10^{-8}	1.0×10^{-6}	1.1×10^{-7}	5.5×10^{-8}	1.2×10^{-6}

4.2.2. Ablation Study

To evaluate the contribution of each AES component, four algorithmic variants were constructed on the 16D-CSTR in addition to a greedy control group, V0, with subsequent variants successively incorporating each core module (Tables 6 and 7). $\checkmark$ indicates that the corresponding mechanism is enabled, and $\times$ indicates that it is disabled.

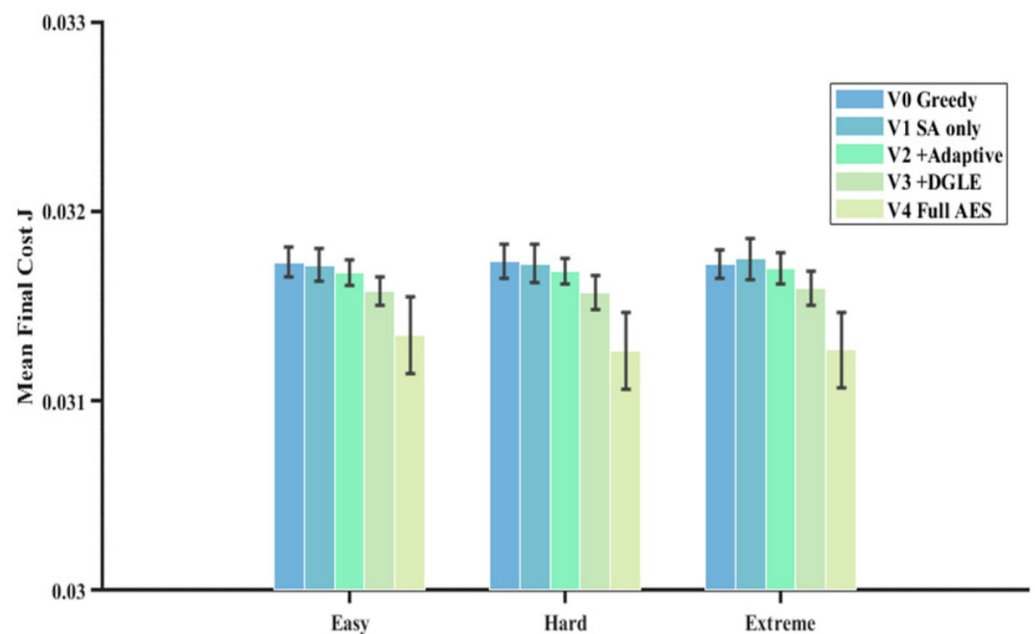
Table 6. AES ablation-experiment settings.

Variant	Adaptive Step Size	DGLE	Confidence Penalty	Local Refinement
V0 (Greedy)	×	×	×	×
V1 (SA only)	×	×	×	×
V2 (+Adaptive)	✓	×	×	×
V3 (+DGLE)	✓	✓	×	×
V4 (Full AES)	✓	✓	✓	✓

Table 7. Ablation results across three initial-point difficulty levels (50 seeds per level; mean $\pm$ std).

Variant	Easy J ($\times 10^{-3}$)	Easy Improv. (%)	Hard J ($\times 10^{-3}$)	Hard Improv. (%)	Extreme J ($\times 10^{-3}$)	Extreme Improv. (%)
V0	31.96 $\pm$ 0.21	15.78 $\pm$ 0.54	32.25 $\pm$ 1.10	23.37 $\pm$ 2.62	31.94 $\pm$ 0.25	29.41 $\pm$ 0.56
V1	31.90 $\pm$ 0.28	15.93 $\pm$ 0.75	31.93 $\pm$ 0.23	24.13 $\pm$ 0.53	31.93 $\pm$ 0.19	29.45 $\pm$ 0.41
V2	31.84 $\pm$ 0.31	16.10 $\pm$ 0.82	31.77 $\pm$ 0.30	24.50 $\pm$ 0.71	31.82 $\pm$ 0.22	29.69 $\pm$ 0.50
V3	31.69 $\pm$ 0.20	16.48 $\pm$ 0.52	31.63 $\pm$ 0.26	24.84 $\pm$ 0.62	31.70 $\pm$ 0.22	29.96 $\pm$ 0.48
V4	31.53 $\pm$ 0.31	16.91 $\pm$ 0.80	31.51 $\pm$ 0.32	25.12 $\pm$ 0.77	31.50 $\pm$ 0.30	30.40 $\pm$ 0.67

Figure 6 shows the same ordering $V4 < V3 < V2 < V1 < V0$ in mean cost at all three levels over 50 seeds. The full AES lowers the AES-stage cost relative to the greedy variant by $0.43, 0.74$ and 0.45×10^{-3} at the easy, hard and extreme levels, winning in 90%, 86% and 88% of the paired runs. Annealing-style acceptance alone gives no significant gain at any level, the adaptive module is significant under hard and extreme starts, DGLE is significant at all levels, and the confidence-weighted penalty with local refinement is significant under easy and extreme starts.

**Figure 6.** Results of AES module ablation experiments under three types of initial points.

4.2.3. Baseline Comparison

AES-RTO was compared against three alternative optimization strategies: Multi-Start SQP [38], which relies on local gradient information from multiple starting points; Bayesian optimization [29], which performs a global search using a Gaussian-process surrogate; and an artificial neural network (ANN) surrogate combined with local fmincon refinement [22].

All methods were initialized from the same post-perturbation cost, $J_0 = 0.03795$, and each method was executed over 50 independent runs. The convergence tolerance of all fmincon-based stages was set to $\text{TolFun} = 10^{-10}$.

Bayesian optimization achieved the lowest final cost, 0.02819, but required 26.37 ± 0.69 s. AES-RTO reached 0.02876 in 4.08 ± 0.38 s; its final cost was 2.02% higher, whereas its runtime was approximately 84.53% lower. Multi-Start SQP was the fastest method, but its improvement was limited to 14.62%, which is consistent with repeated convergence to local optima. The ANN-surrogate method reached a higher final cost of 0.03023, likely because the limited training set introduced surrogate-model error. Which method is preferable depends on how long the new economic conditions persist. If the plant stays at the post-perturbation cost J_0 until method m returns after time t_m and at the delivered cost J_m afterwards, the cumulative cost over a persistence time H is $C_{in}(H) = J_0 t_m + J_m(H - t_m)$, $H \geq t_m$. Bayesian optimization yields the lower cumulative cost only if the conditions persist for longer than roughly 6.4 min, whereas AES-RTO yields a lower cumulative cost than the ANN surrogate and Multi-Start SQP for H above about 14 s and 9 s, respectively. AES-RTO also shows a larger run-to-run variability in final cost than the other methods (Figure 7). The total time in Table 8 is the online computation time, which includes the inverse mapping, the feasibility screening and, when triggered, the feasible-neighbor repair; the offline preparation of the 16D case study, performed once and reused across all re-optimization cycles, required 108.2 s.

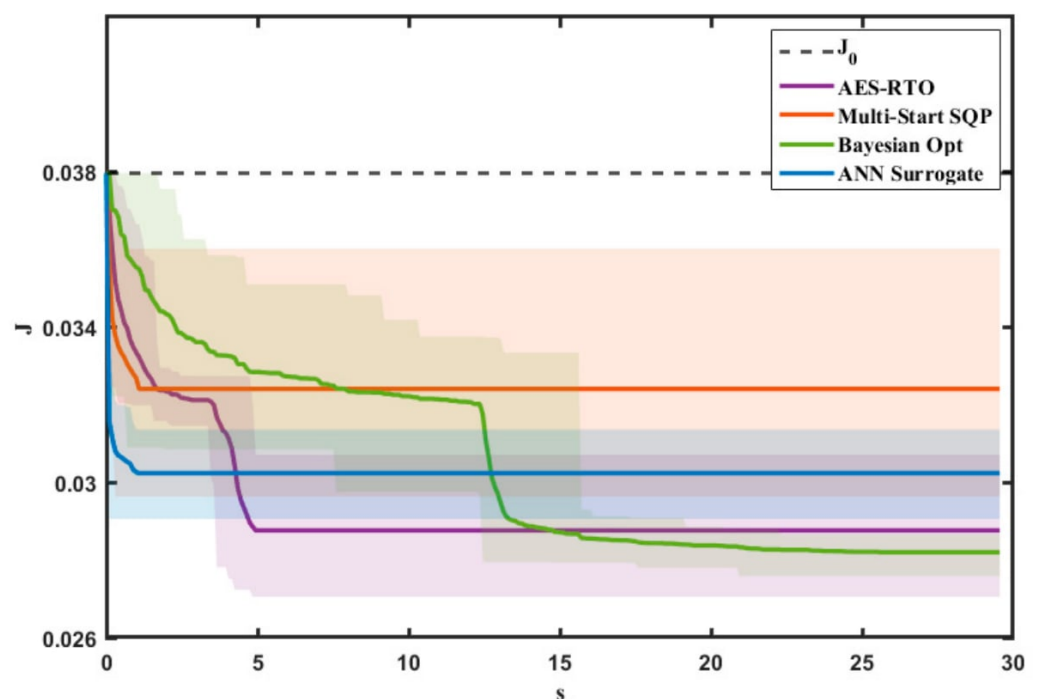


Figure 7. Convergence performance comparison of the four methods.

Table 8. Baseline comparison on the 16D-CSTR (50 runs).

Method	Final Cost J (10^{-3})	Improvement (%)	Total Time (s)
AES-RTO	28.76 ± 1.33	24.22 ± 3.50	4.08 ± 0.38
Multi-Start SQP	32.40 ± 1.07	14.62 ± 2.82	1.11 ± 0.01
Bayesian Opt	28.19 ± 0.25	25.72 ± 0.66	26.37 ± 0.69
ANN Surrogate	30.23 ± 0.54	20.34 ± 1.42	2.13 ± 0.34

4.3. Tennessee Eastman Process

The TEP is a widely used industrial benchmark comprising a reactor, condenser, vapor–liquid separator, stripper, and recycle compressor [39], as shown in Figure 8. In this process, four reactants, A, C, D, and E, react in the gas phase to form the target products G and H, along with an inert component B and a byproduct F. Twelve manipulated variables, XMV₁–XMV₁₂, corresponding to valve-position percentages, were considered, giving the control vector

$$\mathbf{u}_{12} = [u_1, u_2, \dots, u_{12}]^T \in \mathbb{R}^{12} \quad (35)$$

where u_1 – u_{12} correspond, respectively, to the D feed, E feed, A feed, A/C mixed feed, recycle, purge, separator liquid-level valve, stripper liquid-level valve, steam valve, reactor cooling water, condenser cooling water, and agitator speed. Except for the agitator, whose lower bound is 50%, the remaining valve positions range from 0% to 100%. The Mode-1 nominal operating point was adopted as the linearization center [40].

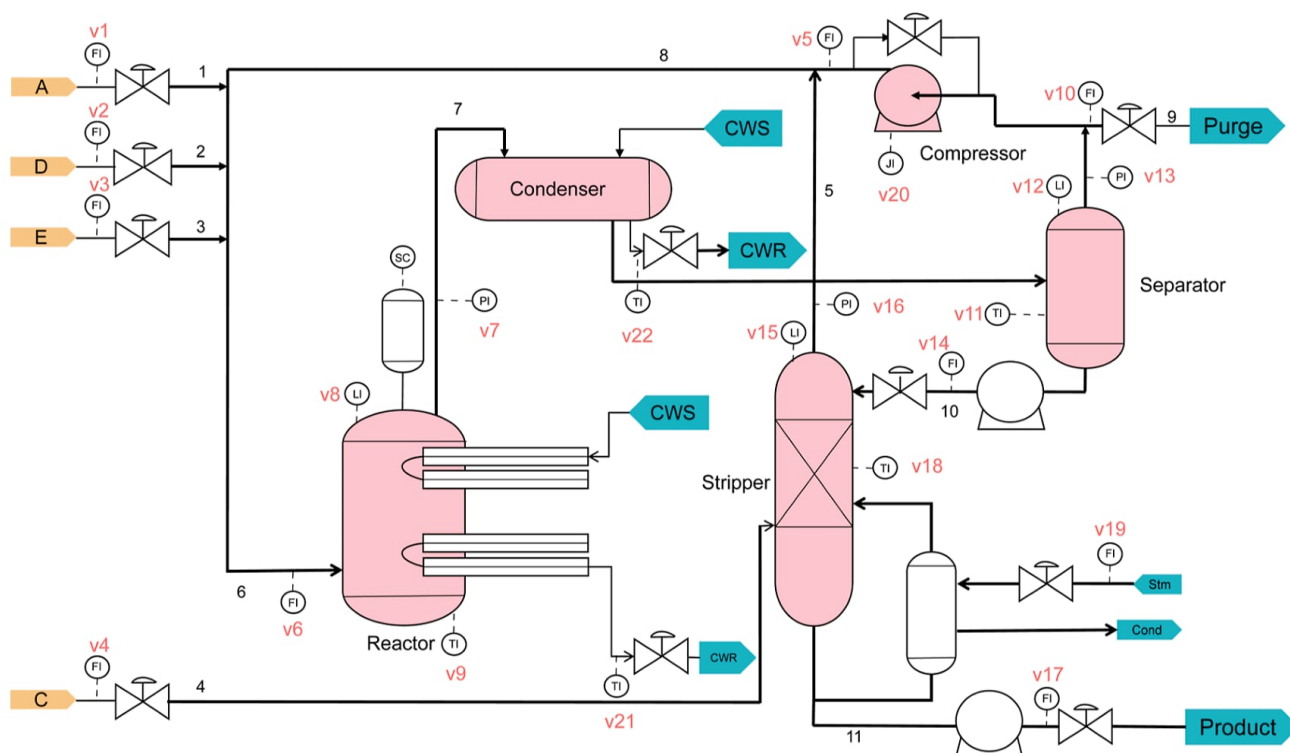


Figure 8. Tennessee Eastman Process flowchart.

Because repeated calls to the full TEP dynamic simulator within the optimization loop are computationally prohibitive, a steady-state surrogate model was constructed around the nominal operating condition. The TEP results therefore refer to this surrogate model and have not been verified with the full dynamic simulator. With the normalized deviation defined as

$$\delta_i = \frac{u_i - u_i^{\text{nom}}}{u_i^{\text{max}} - u_i^{\text{min}}}, \quad i = 1, \dots, 12 \quad (36)$$

the key steady-state outputs were approximated using a first-order expansion supplemented by a small number of second-order correction terms, of the representative form

$$\begin{aligned} P_R &= 2705 - 12.5\delta_6 + 8.3\delta_5 + 6.1\delta_4 + 4.2\delta_3 - 3.8\delta_{10} - 0.5\delta_6^2 \text{ [kPa]} \\ T_R &= 120.4 - 8.2\delta_{10} + 3.1\delta_4 + 2.5\delta_3 - 1.2\delta_5 + 0.8\frac{P_R - 2705}{100} - 0.3\delta_{10}^2 \text{ [}^\circ\text{C]} \\ x_G &= 53.72 + 0.85(T_R - 120.4) - 12.5\frac{u_1 - u_1^{\text{nom}}}{100} + 9.3\frac{u_2 - u_2^{\text{nom}}}{100} + 0.3\theta_1 \text{ [mol\%]} \end{aligned} \quad (37)$$

Compared with the CSTR cases, TEP contains stronger interactions among the manipulated variables. In particular, reactor pressure and temperature are directly coupled through the cross-term in Equation (37), and product composition depends jointly on reactor temperature and the D/E feed split. Therefore, the optimizer must coordinate several variables rather than follow a single dominant search direction. Constraints on reactor pressure, reactor temperature, unit levels, and the G/H product ratio were incorporated into the objective function as quadratic penalty terms. In the TEP case, the economic parameters used in the local refinement of the training samples are perturbed around the post-change values.

The TEP economic objective function represents the total hourly operating cost, decomposed into four components:

$$J_E(\mathbf{u}_{12}) = C_{\text{purge}} + C_{\text{prod}} + C_{\text{cmp}} + C_{\text{stm}} \quad (38)$$

With $\kappa = 1000/22.4$ denoting the conversion factor from kscmh to kmol/h, the individual cost terms are given by

$$\begin{aligned} C_{\text{purge}} &= \kappa \theta_1 \sum_{j \in \{A, C, D, E, F, G, H\}} c_j F_{j, \text{purge}} \\ C_{\text{prod}} &= \kappa \theta_2 \left(c_D F_{D, \text{prod}} + c_E F_{E, \text{prod}} + c_F F_{F, \text{prod}} \right) \\ C_{\text{cmp}} &= c_{\text{cmp}} \theta_3 W_{\text{cmp}} \\ C_{\text{stm}} &= c_{\text{stm}} \theta_2 F_{\text{stm}} \end{aligned} \quad (39)$$

where the per-component molar costs $c_A = 2.206$, $c_C = 6.177$, $c_D = 22.06$, $c_E = 14.56$, $c_F = 17.89$, $c_G = 30.44$, and $c_H = 22.94$. And the utility-cost parameters $c_{\text{stm}} = 0.0318$ \$/kg and $c_{\text{cmp}} = 0.0536$ \$/kW follow standard reference values for this benchmark. The economic weight vector $\theta = [\theta_1, \theta_2, \theta_3]^\top$ scales the purge loss, product/steam-related cost, and compressor energy cost. The nominal condition uses $\theta^{\text{nom}} = [1.0, 1.0, 1.0]^\top$, and the perturbed condition uses $\theta^{\text{new}} = [1.2, 0.8, 1.1]^\top$, representing a simultaneous increase in feedstock cost, decrease in product value, and increase in compression energy cost.

A two-dimensional latent space is used for this case, consistent with the single-stage CSTR case, since the search trajectory and cost landscape are likewise visualized. The AES layer reduced the post-perturbation cost from 286.81 to 260.21 (Table 9). The subsequent fmincon refinement reduced the cost further to 255.43. Thus, the refinement stage contributed 4.78 cost units, or 15.23% of the total improvement, as shown in Figure 9. This contribution was larger than that observed in the 6D-CSTR case, which is consistent with the stronger nonlinear interactions in TEP. In particular, pressure–temperature coupling and the joint effect of the D/E feed ratio on product composition make final local coordination important. AES rapidly identifies a favorable feasible region, whereas the RTO layer refines the coupled variables near the optimum. These results on the TEP steady-state surrogate show that the two-layer architecture can also be applied to a process whose structure differs from the CSTR cases.

Table 9. TEP optimization results (50 runs).

Scheme	Initial Cost J_0	Final Cost J	Improvement (%)	Time (s)
AES	286.81	260.21 $\pm$ 0.80	9.27 $\pm$ 0.28	0.20 $\pm$ 0.02
AES-RTO	286.81	255.43 $\pm$ 0.00	10.94 $\pm$ 0.00	0.27 $\pm$ 0.03

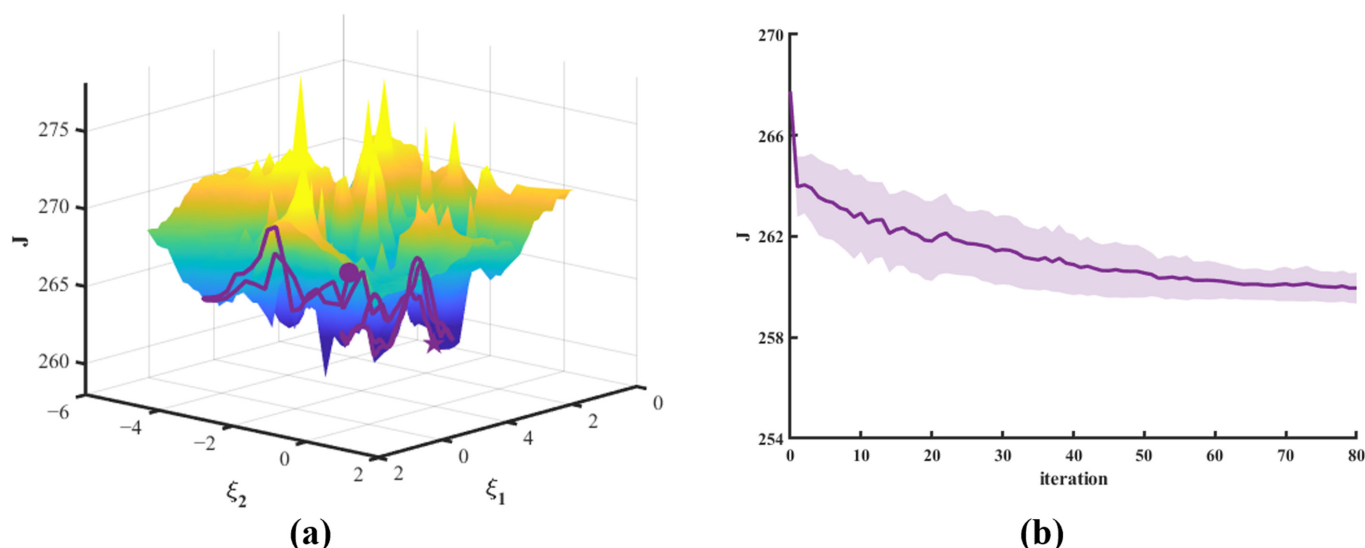


Figure 9. Optimization trajectory of AES in the latent space (a) and convergence curve (b).

5. Conclusions

This paper presents an AES-RTO framework that combines economically weighted manifold learning, adaptive population-based search, and gradient-based RTO refinement for economic optimization of chemical processes with moderately high-dimensional decision spaces. The CSTR and TEP case studies show that AES can identify economically favorable suboptimal regions in the reduced latent space and provide useful warm starts for subsequent local refinement. The inverse-mapping diagnostics further show a final process-feasible rate of 100% on the tested CSTR query sets and full hard-constraint feasibility on the TEP surrogate.

The 16D-CSTR comparison highlights the quality-time trade-off of the proposed framework. Bayesian optimization achieved the lowest final cost when a longer computational budget was available, whereas AES-RTO achieved a better economic improvement than Multi-Start SQP while requiring substantially less time than Bayesian optimization. Therefore, AES-RTO is particularly relevant when an economically useful solution is required within a limited online time budget. In a supplementary scaling study with up to 61 manipulated variables (Appendix C), the cost of the AES stage grew only mildly, whereas the gradient-based refinement stage dominated the online time and the run-to-run variability of the achieved cost increased. The present results demonstrate the algorithmic performance on benchmark models with exact process models and deterministic economic functions. Future work will investigate plant-model mismatch, measurement uncertainty, dynamic disturbances, larger decision spaces, and independent validation using high-fidelity process simulators.

Author Contributions: Conceptualization, B.W.; Methodology, B.W. and Q.J.; Software, B.W. and Y.H.; Validation, B.W. and Y.H.; Writing—Original Draft, B.W. and Y.H.; Writing—Review and Editing, B.W. and Q.J.; Supervision, Q.J.; Funding Acquisition, Q.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Grant Nos. 62673039 and 61673004).

Data Availability Statement: The data presented in this study are available on request from the corresponding author. Since the data are generated from process simulations, the complete model configuration and parameter settings are provided in the main text, and the raw results can be shared upon reasonable request.

Acknowledgments: The authors would like to express their sincere gratitude to their supervisor for their invaluable guidance and support throughout this work. The authors are also deeply grateful to their family and friends for their constant encouragement and companionship.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Table A1. Reconstruction quality of UMAP and PCA latent-space embeddings (single CSTR).

Method	Training Samples (N)	Avg. Reconstruction Error	Cost Correlation	Relative Cost Error (%)
UMAP	2500	0.4415	0.7862	6.82
PCA	2500	1.5424	0.5597	9.96
UMAP	200	0.4934	0.6269	9.10
PCA	200	1.4956	0.7958	6.99

Note: At a limited training-sample size $N = 200$, UMAP exhibits unstable reconstruction fidelity relative to PCA, likely due to insufficient density for reliable neighborhood-graph estimation. As the number of training samples increases to $N = 2500$, UMAP's advantage becomes clear and consistent across all three metrics, whereas PCA's cost correlation degrades from 0.7958 to 0.5597 and its relative cost error increases from 6.99% to 9.96%. This pattern is consistent with the presence of genuine nonlinear curvature in the economically optimal operating-point manifold, which PCA cannot capture regardless of sample size, whereas UMAP benefits from denser local-neighborhood information as more training samples become available. Reconstruction error is reported in normalized operating-point coordinates.

Table A2. AES parameter settings used in the experiments.

(i) Core Iteration and Stagnation Parameters:		
Parameter	Value	Description
T	60	Total iteration budget
λ_0	10	Initial population size
λ_{max}	18	Maximum population size
n_{th}	10	Stagnation count at which the population-expansion sigmoid $g(\cdot)$ is centered, and at which the step-size amplification in $\psi(\cdot)$ activates
s_{smooth}	3	Smoothing width of the sigmoid $g(\cdot)$
Stagnation threshold	15 generations	Consecutive non-improving generations that trigger long-range exploration
Stagnation threshold	10 generations	Consecutive non-improving generations that trigger a soft restart; this counter additionally resets whenever the historical best cost improves, so it is not always identical to non-improving
Early-stop threshold	35 generations	Consecutive non-improving generations that terminate the search
T_{init}	3.0	Initial annealing temperature
τ	45	Decay time constant
σ_{init}	0.30	Baseline step size
(ii) Functional Forms of the Adaptive Schedules:		
Function	Form	
Population-expansion function $g(\cdot)$	$\lambda_k = \lambda_0 + (\lambda_{max} - \lambda_0) \cdot g(s_k)$, where $g(s_k) = \frac{1}{1 + \exp(-(s_k - n_{th})/s_{smooth})}$ and s_k is the current stagnation count	
Step-size decay function $\psi(\cdot)$	$\sigma_k = \sigma_{init} \cdot \exp(-k/\tau)$; multiplied by an additional factor of 1.2 once the stagnation count reaches n_{th}	
Annealing temperature schedule T_k	$T_k = T_{init} \cdot \gamma_k^k$, where the decay base γ_k is re-evaluated at every generation: $\gamma_k = 0.95$ if the search is currently stagnating and $\gamma_k = 0.99$ otherwise	
Long-range exploration mixing weight β_k	$\beta_k = \exp(-k/2\tau)$; applied only to the subset of offspring performing a long-range jump, mixing a locally sampled point with a uniformly sampled point across the full latent-space boundary: $\xi_{trial} = (1 - \beta_k)\xi_{local} + \beta_k\xi_{rand}$	

Table A2. Cont.

(iii) Latent-Space Dimension, UMAP Hyperparameters, and Inverse-Mapping KNN Parameter by Case Study:			
Case study	UMAP n_neighbors	UMAP min_dist	Inverse-mapping KNN k
Single-stage CSTR (6D)	20	0.10	15
Cascade CSTR (16D)	25	0.08	20
TEP (12D)	25	0.08	5

Table A3. One-factor-at-a-time sensitivity analysis of key AES parameters (16D cascade CSTR, 20 runs).

Parameter	Value	Final Cost J (10^{-3})	Improvement (%)	Time (s)
σ_{init}	0.15	34.30 ± 0.96	9.61 ± 2.52	3.54 ± 0.09
σ_{init}	0.30 (baseline)	34.49 ± 0.87	9.12 ± 2.28	3.58 ± 0.12
σ_{init}	0.45	34.72 ± 0.84	8.50 ± 2.22	3.57 ± 0.09
λ_{max}	10	34.48 ± 0.92	9.13 ± 2.43	3.54 ± 0.12
λ_{max}	18 (baseline)	34.49 ± 0.87	9.12 ± 2.28	3.57 ± 0.17
λ_{max}	30	34.06 ± 0.43	10.25 ± 1.13	3.62 ± 0.11
Stagnation thr.	5	34.48 ± 0.83	9.13 ± 2.19	3.57 ± 0.12
Stagnation thr.	15 (baseline)	34.49 ± 0.87	9.12 ± 2.28	3.59 ± 0.13
Stagnation thr.	30	34.49 ± 0.87	9.12 ± 2.28	3.59 ± 0.12
T_{init}	1.0	34.58 ± 0.83	8.86 ± 2.20	3.54 ± 0.11
T_{init}	3.0 (baseline)	34.49 ± 0.87	9.12 ± 2.28	3.56 ± 0.12
T_{init}	6.0	34.49 ± 0.87	9.12 ± 2.28	3.56 ± 0.12
τ	20	34.42 ± 0.83	9.30 ± 2.20	3.54 ± 0.11
τ	45 (baseline)	34.49 ± 0.87	9.12 ± 2.28	3.56 ± 0.19
τ	90	34.45 ± 0.69	9.21 ± 1.81	3.53 ± 0.09

Summary: AES-RTO is broadly robust to all five tested parameters within the examined ranges. Increasing λ_{max} from 18 to 30 improves both the mean improvement and consistency at a negligible time cost. The exploration temperature T_{init} exhibits a saturation effect: once $T_{init} \geq 3$, essentially all uphill candidates are unconditionally accepted, so further increases produce no additional effect. The stagnation threshold and decay time constant τ have negligible effect within the tested ranges.

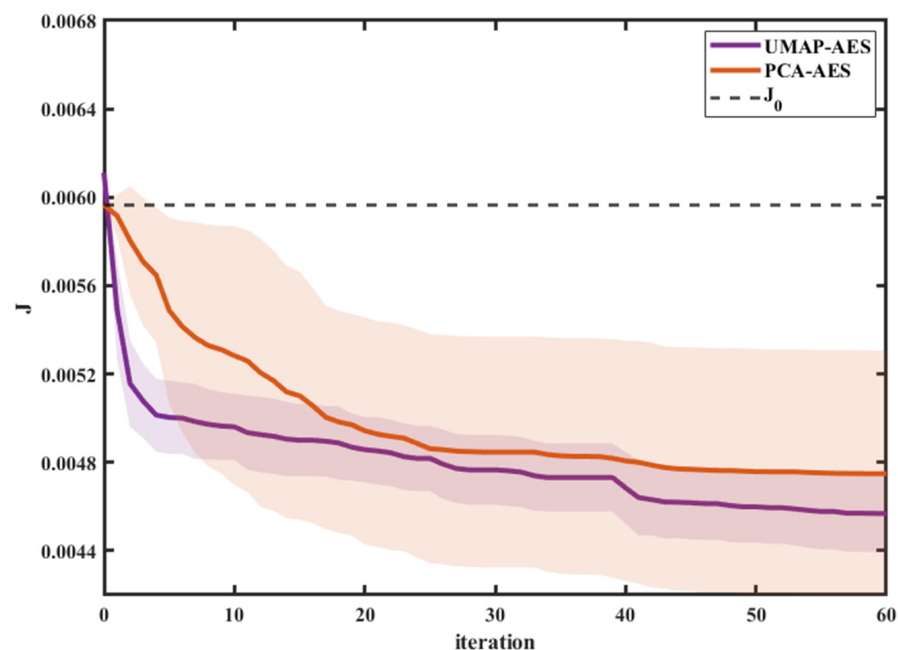


Figure A1. Comparison of convergence of UMAP-AES and PCA-AES on single CSTR.

Appendix B

Table A4. Process-feasibility diagnostics of the UMAP-KNN inverse mapping for the CSTR cases.

Case	Source	N	Direct Feasible	Final Feasible	Mean r_{eq}	P95 r_{eq}	Max r_{eq}	Map/Screen (ms)
6D	AES offspring	5000	100%	100%	1.16×10^{-11}	3.27×10^{-11}	--	0.130/13.708
6D	Uniform latent	1000	100%	100%	5.60×10^{-12}	1.04×10^{-11}	--	0.133/13.833
6D	Boundary latent	500	100%	100%	1.03×10^{-11}	1.04×10^{-11}	--	0.136/13.909
16D	AES offspring	5000	98.88%	100%	6.29×10^{-8}	2.15×10^{-7}	2.23×10^{-5}	0.166/25.527
16D	Uniform latent	1000	98.30%	100%	7.94×10^{-8}	1.86×10^{-7}	1.84×10^{-5}	0.163/25.448
16D	Boundary latent	500	99.80%	100%	3.87×10^{-8}	1.59×10^{-7}	1.89×10^{-6}	0.161/25.335

Directly feasible denotes feasibility after bound projection and before nearest feasible-neighbor repair. The maximum inequality-constraint violation was zero for all CSTR query sets. Neighbor-filter fallback was not activated. Repair time, conditional on activation, was 265–277 ms for the 16D case.

Table A5. Local-surrogate constraint-feasibility and product-quality diagnostics of the UMAP-KNN inverse mapping for the TEP case.

Source	N	Hard Feasible	Max Violation	Mean $ \Delta x_G $	P95 $ \Delta x_G $	P95 $ \Delta(G/H) $	Quality Penalty	Map/Screen (ms)
AES offspring	5000	100%	0	0.0393	0.1393	0.00709	0.1429	0.168/0.090
Uniform latent	1000	100%	0	0.0410	0.1608	0.00815	0.1605	0.165/0.086
Boundary latent	500	100%	0	0.0371	0.1621	0.00820	0.1400	0.167/0.087

TEP feasibility is evaluated using the local surrogate model. Product-composition deviations are reported as soft-quality diagnostics because the current objective penalizes composition deviations rather than imposing an explicit hard G/H-ratio tolerance.

Appendix C

To examine how the computational cost scales with the number of manipulated variables, the cascade CSTR of Section 4.2 was extended to N stages with $m = 6 + 5(N - 1)$ manipulated variables and $2N$ steady-state equality constraints. The steady-state equations are solved inside every cost evaluation, so the optimizers handle only box constraints. Table A6 reports 10 independent runs per case with a three-dimensional latent space. The evaluation cap of the SQP refinement stage was set to 3000 for $m \leq 26$, where the stage needs on average fewer than 1200 evaluations, and to $200(m + 1)$ for $m > 26$, where no run reached the cap. The number of cost evaluations of the AES stage stays at about $1.0\text{--}1.1 \times 10^3$ and its time grows from 3.2 s to 8.7 s, whereas the time of the gradient-based refinement stage grows from 1.0 s to 31 s and accounts for 78% of the online time at $m = 61$. The mean cost improvement decreases from 21.5% to 16.8%, and its run-to-run variability increases with the dimension. At $m = 61$, latent dimensions of 3, 5, 8, 10 and 15 gave mean improvements of 16.8%, 13.6%, 12.3%, 20.7% and 20.7%, so no monotonic dependence on the latent dimension was observed.

Table A6. Computational scaling of AES-RTO on N-stage cascade CSTRs (10 runs per case; mean $\pm$ std).

m	Eq. Constr.	Latent Dim.	Offline (s)	Online Total (s)	AES Stage (s)	SQP Stage (s)	AES-Stage Evals	Improv. Mean/Median (%)
16	6	3	100.6	4.21 $\pm$ 0.28	3.22	0.99	1055	21.5/20.9
26	10	3	239.8	8.99 $\pm$ 1.49	5.09	3.91	1135	19.7/16.5
41	16	3	492.5	16.35 $\pm$ 2.71	6.61	9.73	1103	16.7/14.2
61	24	3	691.5	39.93 $\pm$ 13.82	8.69	31.24	1113	16.8/12.7
61	24	5	733.9	34.23 $\pm$ 8.07	8.91	25.32	1084	13.6/12.0
61	24	10	705.0	43.43 $\pm$ 13.88	8.31	35.12	1015	20.7/20.6

References

- Zhang, Z.; Li, S. Enhanced reinforcement learning in two-layer economic model predictive control for operation optimization in dynamic environment. *Chem. Eng. Res. Des.* **2023**, *196*, 133–143. [\[CrossRef\]](#)
- Alhazmi, K.; Sarathy, S.M. Dynamic optimizers for complex industrial systems via direct data-driven synthesis. *Commun. Eng.* **2025**, *4*, 25. [\[CrossRef\]](#) [\[PubMed\]](#)
- Ellis, M.; Christofides, P.D. On finite-time and infinite-time cost improvement of economic model predictive control for nonlinear systems. *Automatica* **2014**, *50*, 2561–2569. [\[CrossRef\]](#)
- Ellis, M.; Durand, H.; Christofides, P.D. A tutorial review of economic model predictive control methods. *J. Process Control* **2014**, *24*, 1156–1178. [\[CrossRef\]](#)
- Xu, J. Efficiency-oriented model predictive control: A novel MPC strategy to optimize the global process performance. *Sensors* **2024**, *24*, 5732. [\[CrossRef\]](#) [\[PubMed\]](#)
- Wu, J.; Liu, F. Stable nonlinear model predictive control with a changing economic criterion. *Int. J. Control.* **2024**, *97*, 1488–1499. [\[CrossRef\]](#)
- Huang, M.; Zheng, Y.; Li, S. Distributed economic model predictive control with pseudo-steady state modifier adaptation for an industrial fluid catalytic cracking unit. *Chem. Eng. Res. Des.* **2022**, *180*, 379–390. [\[CrossRef\]](#)
- Patrón, G.D.; Toffolo, K.; Ricardez-Sandoval, L. Economic model predictive control for packed bed chemical looping combustion. *Chem. Eng. Process. Process Intensif.* **2024**, *198*, 109731. [\[CrossRef\]](#)
- Biegler, L.T. Multi-level optimization strategies for large-scale nonlinear process systems. *Comput. Chem. Eng.* **2024**, *185*, 108657. [\[CrossRef\]](#)
- Wu, Z.; Rincon, D.; Christofides, P.D. Real-time adaptive machine-learning-based predictive control of nonlinear processes. *Ind. Eng. Chem. Res.* **2020**, *59*, 2275–2290. [\[CrossRef\]](#)
- Kaiser, E.; Kutz, J.N.; Brunton, S.L. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proc. R. Soc. A Math. Phys. Eng. Sci.* **2018**, *474*, 20180335. [\[CrossRef\]](#) [\[PubMed\]](#)
- Vaupel, Y.; Hamacher, N.C.; Caspari, A.; Mhamdi, A.; Kevrekidis, I.G.; Mitsos, A. Accelerating nonlinear model predictive control through machine learning. *J. Process. Control.* **2020**, *92*, 261–270. [\[CrossRef\]](#)
- Krishnamoorthy, D. A sensitivity-based data augmentation framework for model predictive control policy approximation. *IEEE Trans. Autom. Control.* **2022**, *67*, 6090–6097. [\[CrossRef\]](#)
- Patrón, G.D.; Ricardez-Sandoval, L. Directional modifier adaptation based on input selection for real-time optimization. *Comput. Chem. Eng.* **2023**, *177*, 108351. [\[CrossRef\]](#)
- Bottari, A. Real-time optimization modifier adaptation approach using quadratic approximation of the plant-model mismatch function. *J. Process. Control.* **2024**, *136*, 103180. [\[CrossRef\]](#)
- Zhang, D.; Li, X.; Wang, K.; Shao, Z. A globally convergent composite-step trust-region framework for real-time optimization. *AIChE J.* **2024**, *70*, e18300. [\[CrossRef\]](#)
- Kadam, J.V.; Ray, A.; Bhushan, S. A two-level strategy of integrated dynamic optimization and control of industrial processes—A case study. *Comput. Aided Chem. Eng.* **2002**, *10*, 511–516. [\[CrossRef\]](#)
- Würth, L.; Hannemann, R.; Marquardt, W. A two-layer architecture for economically optimal process control and operation. *J. Process. Control.* **2011**, *21*, 311–321. [\[CrossRef\]](#)
- Yang, D.; Liu, T.; Song, D.; Zhang, X.; Zeng, X. A real time multi-objective optimization guided-MPC strategy for power-split hybrid electric bus based on velocity prediction. *Energy* **2023**, *276*, 127583. [\[CrossRef\]](#)
- Healy, J.; McInnes, L. Uniform manifold approximation and projection. *Nat. Rev. Methods Prim.* **2024**, *4*, 82. [\[CrossRef\]](#)
- Kim, S.; Lee, M.; Kim, D.; Yoon, S.; Lee, D. AutoML-driven soft sensors for real-time monitoring of amino acids in mammalian perfusion cultures. *Biotechnol. Bioeng.* **2025**, *122*, 3123–3138. [\[CrossRef\]](#) [\[PubMed\]](#)

22. Hassanpour, H.; Corbett, B.; Mhaskar, P. Artificial neural network-based model predictive control using correlated data. *Ind. Eng. Chem. Res.* **2022**, *61*, 3075–3090. [CrossRef]
23. Marra, L.; Semeraro, O.; Mathelin, L.; Meilán-Vila, A.; Discetti, S. Latent-space nonlinear model predictive control for partially observable systems. *Proc. R. Soc. A Math. Phys. Eng. Sci.* **2026**, *482*, 20251027. [CrossRef]
24. Zhao, T.; Zheng, Y.; Gong, J.; Wu, Z. Machine learning-based reduced-order modeling and predictive control of nonlinear processes. *Chem. Eng. Res. Des.* **2022**, *179*, 435–451. [CrossRef]
25. Luo, L.; Zhou, Y.; Zhou, Z.; Zhou, C.; Ji, X.; Liu, B.; He, G. Online optimization of petrochemical process via case-based reasoning and conditional mutual information. *Chem. Eng. Res. Des.* **2024**, *207*, 380–391. [CrossRef]
26. Lei, N.; Zhang, H.; Pang, Z.; Hu, J.; Chen, H.; Hu, Z.; Wang, Z. Physics-informed surrogate-assisted co-optimization of fuel cell electric vehicle powertrain and energy management for long-haul heavy-duty applications. *Appl. Energy* **2026**, *414*, 127828. [CrossRef]
27. Lei, N.; Zhang, H.; Wang, H.; Wang, Z. An improved co-optimization of component sizing and energy management for hybrid powertrains interacting with high-fidelity model. *IEEE Trans. Veh. Technol.* **2023**, *72*, 15585–15596. [CrossRef]
28. Spall, J. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Trans. Autom. Control* **1992**, *37*, 332–341. [CrossRef]
29. Gui, Y.; Zhan, D.; Li, T. Taking another step: A simple approach to high-dimensional Bayesian optimization. *Inf. Sci.* **2024**, *679*, 121056. [CrossRef]
30. Hvarfner, C.; Hellsten, E.O.; Nardi, L. Vanilla Bayesian optimization performs great in high dimensions. In Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria, 21–27 July 2024; Volume 235, pp. 20793–20817. Available online: <https://proceedings.mlr.press/v235/hvarfner24a.html> (accessed on 20 May 2026).
31. Li, Z.; Wu, W.; Zhang, Q.; Cai, X. Exploring high-dimensional optimization by sparse and low-rank evolution strategy. *Swarm Evol. Comput.* **2025**, *92*, 101828. [CrossRef]
32. Fu, W.-Y. Accelerated high-dimensional global optimization: A particle swarm optimizer incorporating homogeneous learning and autophagy mechanisms. *Inf. Sci.* **2023**, *645*, 119573. [CrossRef]
33. Lei, N.; Zhang, H.; Li, R.; Yu, J.; Wang, H.; Wang, Z. Physics-informed data-driven modeling approach for commuting-oriented hybrid powertrain optimization. *Energy Convers. Manag.* **2024**, *299*, 117814. [CrossRef]
34. Lei, N.; Zhang, H.; Wang, H.; Hu, Z.; Chen, H.; Hu, J.; Wang, Z. Theory-constrained neural network with modular interpretability for fuel cell vehicle modeling. *IEEE Trans. Veh. Technol.* **2025**, *74*, 8907–8920. [CrossRef]
35. Mokhtar, M.A.; Fathy, M.; Dahab, Y.A.; Sayed, E.A. A dual enhanced stochastic gradient descent method with dynamic momentum and step size adaptation for improved optimization performance. *Sci. Rep.* **2025**, *15*, 40389. [CrossRef] [PubMed]
36. Lange, R.T.; Schaul, T.; Chen, Y.; Zahavy, T.; Dalibard, V.; Lu, C.; Singh, S.; Flennerhag, S. Discovering evolution strategies via meta-black-box optimization. In Proceedings of the International Conference on Learning Representations (ICLR) 2023, Kigali, Rwanda, 1–5 May 2023. Available online: <https://openreview.net/forum?id=mFDU0fP3EQH> (accessed on 26 May 2026).
37. Abedin, M.; Tabata, K.; Matsumura, Y.; Komatsuzaki, T. Multi-armed bandit algorithm for sequential experiments of molecular properties with dynamic feature selection. *J. Chem. Phys.* **2024**, *161*, 014115. [CrossRef] [PubMed]
38. Li, X.-B.; Yang, L.-C. Study of constrained nonlinear thrust allocation in ship application based on optimization and SOM. *Ocean Eng.* **2019**, *191*, 106491. [CrossRef]
39. Mustafa, F.E.; Ahmed, I.; Basit, A.; Alqahtani, M.; Khalid, M. An adaptive metaheuristic optimization approach for Tennessee Eastman process for an industrial fault-tolerant control system. *PLoS ONE* **2024**, *19*, e0296471. [CrossRef] [PubMed]
40. Ricker, N.L.; Lee, J. Nonlinear model predictive control of the Tennessee Eastman challenge process. *Comput. Chem. Eng.* **1995**, *19*, 961–981. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.