Ensemble Kalman–Guided Model Predictive Path Integral Control for Spatially Localized Suppression of Extremes in Chaotic Geophysical Flows

Haru Kuroki[1], Kazumune Hashimoto[1], Yuki Uehara[1], Yohei Sawada[2], Duc Le[2], and Masashi Minamide[2]

[1]The University of Osaka
[2]The University of Tokyo

Correspondence: Kazumune Hashimoto (hashimoto@eei.eng.osaka-u.ac.jp)

Abstract. The possibility of influencing extreme weather phenomena has been discussed for decades; however, it remains far from operational practice, and there is still no established framework for designing small, spatially localized perturbations that can reliably steer chaotic geophysical flows. In this study, we propose a hybrid control method, termed ensemble-Kalman-guided model predictive path integral control (EKG-MPPI), which combines ensemble Kalman control (EnKC) with model predictive path integral (MPPI) control. Within a control simulation experiment framework, an ensemble Kalman filter is first used for state estimation, after which EnKC computes a candidate perturbation by treating the control objective as a pseudo-observation. An adaptive thresholding procedure then enforces spatial sparsity, so that the EnKC perturbation identifies candidate actuator locations and their nominal amplitudes. This information is embedded into the mean and covariance of Gaussian proposal distributions for MPPI, which subsequently refines the perturbation through sampling-based optimization with nonlinear rollouts, without linearizing the dynamics or computing gradients. Numerical experiments with the Lorenz–96 model and the surface quasi-geostrophic (SQG) model demonstrate that EKG-MPPI can suppress extremes in state variables and regional wind speed more effectively than EnKC alone, while using comparable or smaller control inputs. These results highlight EKG-MPPI as a promising building block for simulation-based assessment of localized intervention strategies in geophysical flows.

1 Introduction

The idea of deliberately influencing extreme weather phenomena, such as tropical cyclones (TCs), has been discussed for decades. However, it remains far from operational practice and there is still no established framework for designing small, spatially localized interventions with predictable effects. For example, Project STORMFURY attempted to weaken maximum wind speeds by artificially inducing convection around the TC eyewall, but its lack of effectiveness was reported by Willoughby et al. (1985). More recently, several studies have explored potential intervention mechanisms primarily in numerical modeling settings: based on simulations by Zhang et al. (2006), Cotton et al. (2007) discussed the possibility that mineral dust injection could suppress TC development, and Saharan dust has been shown to exert a strong influence on TCs. Furthermore, Latham et al. (2012) investigated TC weakening via sea-surface-temperature reduction, and Jacobson and Kempton (2014) demon-

strated in simulations that arrays of offshore wind turbines can suppress near-surface wind speeds. A comprehensive review of these proposed approaches, together with feasibility and governance considerations, is provided by Miller et al. (2023). While these studies identify candidate actuators such as aerosols, mineral dust, and wind turbines, achieving concrete objectives (e.g., a substantial reduction in maximum wind speed) requires more than selecting an actuator type. It also calls for a mathematical framework that systematically optimizes *where*, *when*, and *how strongly* perturbations should be applied. This need is particularly acute because realistic anthropogenic influences are inherently local, weak, and intermittent. Therefore, it is crucial to devise methods that can efficiently identify "small perturbations" that can nevertheless meaningfully steer the evolution of chaotic geophysical flows, at least in simulation-based assessment settings.

An early step in this direction was taken by Henderson et al. (2005), who applied four-dimensional variational data assimilation (4D-Var), widely used in numerical weather prediction, to derive optimal *initial* perturbations for mitigating damage from tropical cyclones. This approach assumes that perturbations are applied only at the initial time. In contrast, Miyoshi and Sun (2022) proposed a framework in which small perturbations are applied continuously and adaptively. This concept, referred to as control simulation experiments (CSEs), has subsequently been employed and extended in later studies, including Kawasaki and Kotsuki (2024) and Ouyang et al. (2023), and has emerged as a practical approach for systematically evaluating the feasibility and potential side effects of candidate intervention strategies.

More recently, Sawada (2024b) introduced ensemble Kalman control (EnKC), which leverages the EnKF/EnKS framework to compute control increments by assimilating the control objective as a pseudo–observation, and demonstrated its effectiveness in numerical experiments with the Lorenz–63 model. Building on this, Sawada (2024a) proposed a strategy to reduce the magnitude of EnKC control inputs and to enforce spatial sparsity, with demonstrations in the Lorenz–96 model. While these studies Sawada (2024b, a) indicate that EnKC can identify effective perturbations, its computation relies on an approximate linearization of the dynamics over the prediction horizon, motivating strategies that can more fully exploit nonlinear dynamics.

To address the linearization limitation of EnKC while retaining a derivative-free formulation, we combine EnKC with model predictive path integral (MPPI) control. Like EnKC, MPPI is derivative-free; however, it evaluates candidate perturbations through nonlinear forward rollouts rather than linearized dynamics. At the same time, MPPI can be sample-inefficient when informative prior knowledge about the control distribution is unavailable, particularly in high-dimensional control spaces Power and Berenson (2022). We therefore introduce EnKC-guided MPPI (EKG-MPPI), which uses the sparse perturbations obtained from EnKC to construct an informative sampling distribution for MPPI. In this way, EnKC provides ensemble-based candidate perturbations that encode *where* control is likely to be effective and their nominal *magnitude*, while MPPI refines them via sampling-based optimization under the full nonlinear model without computing gradients. We demonstrate the effectiveness of the proposed method through numerical experiments with the Lorenz–96 model and the surface quasi-geostrophic (SQG) model.

This paper makes three contributions: (i) we propose EKG-MPPI, a hybrid framework that couples EnKC-based sparse actuation proposals with MPPI-based nonlinear refinement; (ii) we provide an explicit algorithmic formulation of EKG-MPPI, including a practical procedure for embedding EnKC-derived actuator locations and amplitudes into Gaussian sampling dis-

tributions for MPPI; and (iii) through numerical experiments with the Lorenz–96 and SQG models, we show that EKG-MPPI suppresses target extremes more strongly than EnKC (and uninformed MPPI) while using comparable or smaller control inputs.

2 Preliminary Knowledge

This section reviews the Ensemble Kalman filter/control and Model Predictive Path Integral control.

2.1 Ensemble Kalman Filter (EnKF)

Let us first review the EnKF. Consider a discrete-time state–space system

$$\mathbf{x}_t = M(\mathbf{x}_{t-1}) + \mathbf{q}_{t-1}, \tag{1}$$

$$\mathbf{y}_t^o = H(\mathbf{x}_t) + \mathbf{r}_t, \tag{2}$$

where $\mathbf{x}_t$ denotes the state vector, M the forecast model, $\mathbf{q}_{t-1}$ the model error or the system noise, $\mathbf{y}_t^o$ the observation vector, H the observation operator, and $\mathbf{r}_t$ the observation error or the measurement noise. At an assimilation time t, the EnKF updates the model forecast by approximately minimizing the quadratic cost function

$$J(\mathbf{x}_t) = \tfrac{1}{2}(\mathbf{x}_t - \bar{\mathbf{x}}_t^b)^\top \mathbf{P}_b^{-1}(\mathbf{x}_t - \bar{\mathbf{x}}_t^b) + \tfrac{1}{2}(\mathbf{y}_t^o - H(\mathbf{x}_t))^\top \mathbf{R}^{-1}(\mathbf{y}_t^o - H(\mathbf{x}_t)), \tag{3}$$

where $\bar{\mathbf{x}}_t^b$ and $\mathbf{P}_b$ are the ensemble mean and background error covariance, and $\mathbf{R}$ is the observation error covariance matrix. The analysis ensemble $\{\mathbf{x}_t^{a(i)}\}_{i=1}^N$ is obtained by applying the EnKF update

$$\mathbf{x}_t^{a(i)} = \mathbf{x}_t^{f(i)} + \mathbf{K}\big(\mathbf{y}_t^{o(i)} - H(\mathbf{x}_t^{f(i)})\big), \tag{4}$$

$$\mathbf{K} = \mathbf{P}_b\mathbf{H}^\top\big(\mathbf{H}\mathbf{P}_b\mathbf{H}^\top + \mathbf{R}\big)^{-1}, \tag{5}$$

where $\mathbf{x}_t^{f(i)}$ and $\mathbf{x}_t^{a(i)}$ denote the ith forecast and analysis ensemble members, respectively, and $\mathbf{H}$ denotes the matrix representation of the (possibly linearized) observation operator H. In practice, the products $\mathbf{P}_b\mathbf{H}^\top$ and $\mathbf{H}\mathbf{P}_b\mathbf{H}^\top$ are computed using ensemble statistics, so that the full covariance matrix $\mathbf{P}_b$ need not be formed explicitly. This ensemble representation enables the EnKF to handle high-dimensional systems efficiently. To mitigate sampling errors arising from the use of a finite ensemble, covariance localization is commonly employed. The localized Kalman gain is written as

$$\mathbf{K} = \rho \circ \mathbf{P}_b\mathbf{H}^\top\big(\rho \circ (\mathbf{H}\mathbf{P}_b\mathbf{H}^\top) + \mathbf{R}\big)^{-1}, \tag{6}$$

where $\circ$ denotes the Schur (element-wise) product and

$$\rho = \exp[-d(i,j)/L] \tag{7}$$

is a smooth correlation function defined in terms of the distance $d(i,j)$ between grid points and a localization length scale L. This localization suppresses spurious long-range correlations, thereby improving the stability and accuracy of the EnKF updates.

Closely related to the EnKF is the ensemble Kalman smoother (EnKS), which extends the filter in time by estimating the state over an assimilation window using both past and future observations. Operationally, the EnKS can be implemented by propagating an ensemble forward with EnKF updates at each observation time and then applying a backward smoothing step that uses ensemble-based cross-covariances between states at different times and the observations. Readers interested in further details of the EnKF/EnKS family are referred to Houtekamer and Zhang (2016) for a comprehensive review.

2.2 Ensemble Kalman Control (EnKC)

EnKC builds directly on the EnKF and EnKS framework to formulate and solve an optimal control problem for various state-space (dynamical) systems. After the EnKF analysis at time t, let $\{\mathbf{x}_t^{a(i)}\}_{i=1}^N$ denote the analysis ensemble, with mean $\bar{\mathbf{x}}_t^a$ and error covariance $\mathbf{P}_a$. Over a prediction horizon T_{EnKC}, EnKC considers the quadratic cost function

$$J_c(\mathbf{x}_t) = \frac{1}{2}\big(\mathbf{x}_t - \bar{\mathbf{x}}_t^a\big)^\top \mathbf{P}_a^{-1}\big(\mathbf{x}_t - \bar{\mathbf{x}}_t^a\big) + \frac{1}{2}\big(\mathbf{r}_{t+T_{\mathrm{EnKC}}} - H_c(\mathbf{x}_{t+T_{\mathrm{EnKC}}})\big)^\top \mathbf{R}_c^{-1}\big(\mathbf{r}_{t+T_{\mathrm{EnKC}}} - H_c(\mathbf{x}_{t+T_{\mathrm{EnKC}}})\big), \tag{8}$$

subject to the model dynamics

$$\mathbf{x}_{k+1} = M(\mathbf{x}_k), \quad k = t, \ldots, t + T_{\mathrm{EnKC}} - 1. \tag{9}$$

Here, $\mathbf{r}_{t+T_{\mathrm{EnKC}}}$ is a prescribed reference (target) vector at time $t + T_{\mathrm{EnKC}}$, H_c maps the state variables to the control criteria (e.g., a regional average or maximum of a physical quantity), $\mathbf{R}_c$ is a user-defined weight (pseudo–observation error covariance), and $\bar{\mathbf{x}}_t^a$ denotes the forcast ensemble mean. The first term penalizes the size of the perturbation to the initial analysis state, while the second term penalizes the mismatch between the predicted future state and the control objective. Assuming that the dynamics over the prediction horizon are approximately linear, the minimizer of J_c can be obtained by applying an EnKS. In this formulation, the reference vector $\mathbf{r}_{t+T_{\mathrm{EnKC}}}$ is assimilated as a "pseudo–observation" with error covariance $\mathbf{R}_c$. The resulting EnKS analysis at time t,

$$\mathbf{x}_t^c = \bar{\mathbf{x}}_t^a + K\big(\mathbf{r}_{t+T_{\mathrm{EnKC}}} - H_c(\mathbf{x}_{t+T_{\mathrm{EnKC}}}^a)\big), \tag{10}$$

yields the optimal perturbation $\mathbf{x}_t^c - \bar{\mathbf{x}}_t^a$ to be applied to the system. The Kalman gain K is computed from the cross–covariance between the analysis ensemble at time t and the ensemble prediction at $t + T_{\mathrm{EnKC}}$, and from the covariance of the projected prediction, in direct analogy with standard EnKF/EnKS formulations.

The EnKC algorithm can be summarized as follows:

1. Apply EnKF with real observations to obtain the analysis ensemble at time t.
2. From the analysis ensemble, perform ensemble forecasting up to $t + T_{\mathrm{EnKC}}$ and project the predicted states onto the control criteria via H_c.
3. Run EnKS, assimilating $\mathbf{r}_{t+T_{\mathrm{EnKC}}}$ as a pseudo–observation with error covariance $\mathbf{R}_c$, and obtain the perturbation $\mathbf{x}_t^c - \bar{\mathbf{x}}_t^a$.
4. Add this perturbation to the real system and to all analysis ensemble members, thereby updating the controlled "nature" and its ensemble representation.

5. Propagate the updated ensemble to the next data assimilation time. The resulting forecast serves as the new background for the next EnKF cycle, thus completing the loop and returning to Step 1.

In this way, EnKC integrates ensemble data assimilation and model predictive control by treating the control objective as a pseudo–observation within an EnKS framework, and by interpreting the resulting analysis increment as the optimal small perturbation to steer the system toward the desired future state. The control perturbation estimated by EnKC (Sawada (2024a)), $\mathbf{x}_t^c - \bar{\mathbf{x}}_t^a$, has the same dimension as the full model state. In practical applications such as weather modification, however, it is unrealistic to apply perturbations to all state variables at every control step. It is therefore desirable to enforce *spatial sparsity* so that only a limited number of grid points (or regions) are actively perturbed, which naturally aligns with the interpretation of actuators placed at specific locations. Ideally, such sparsity can be promoted by augmenting the EnKC cost function with an ℓ_0-norm penalty:

$$J_c(\mathbf{x}_t) = \frac{1}{2}(\mathbf{x}_t - \bar{\mathbf{x}}_t^a)^\top \mathbf{P}_a^{-1}(\mathbf{x}_t - \bar{\mathbf{x}}_t^a) + \frac{1}{2}(\mathbf{r}_{t+T_{\mathrm{EnKC}}} - H_c(\mathbf{x}_{t+T_{\mathrm{EnKC}}}))^\top \mathbf{R}_c^{-1}(\mathbf{r}_{t+T_{\mathrm{EnKC}}} - H_c(\mathbf{x}_{t+T_{\mathrm{EnKC}}})) + \lambda_{\mathrm{s}} \|\mathbf{x}_t - \bar{\mathbf{x}}_t^a\|_0, \quad (11)$$

where $\|\cdot\|_0$ counts the number of nonzero elements (i.e., ℓ_0-norm) and λ_{s} controls the strength of the sparsity constraint. However, direct minimization of this objective is computationally demanding for high-dimensional systems, so an empirical yet effective strategy is adopted.

Following the idea of Schneider et al. (2022), we impose sparsity on the EnKC-estimated perturbation by applying a thresholding operator to its components. Specifically, after computing the standard EnKC update, we apply a thresholding operator $T(\theta)$ to each entry of $\mathbf{x}_t^c - \bar{\mathbf{x}}_t^a$ so that small-magnitude components are discarded as noise and only sufficiently large perturbations are retained:

$$T(\theta) = \begin{cases} 0, & \text{if } |\theta| < \sqrt{2\lambda_{\mathrm{s}}}, \\ \theta, & \text{otherwise.} \end{cases} \quad (12)$$

While Schneider et al. (2022) treated $\sqrt{2\lambda_{\mathrm{s}}}$ as a fixed hyperparameter, the present study adopts an *adaptive* formulation in which the threshold depends on the amplitude of the EnKC-derived perturbation at each control step. More precisely, the threshold parameter is updated as

$$\sqrt{2\lambda_{\mathrm{s}}} = \Lambda \max |\mathbf{x}_t^c - \bar{\mathbf{x}}_t^a| . \quad (13)$$

where Λ is a user-specified coefficient that controls the enforced sparsity. This adaptive thresholding allows the sparsity level to automatically adjust to the magnitude of the EnKC-derived perturbations, thereby concentrating actuator effort on only the most influential grid points. In particular, choosing $\Lambda = 1$ retains only the grid point with the largest perturbation, which is consistent with the actuator-placement interpretation in which localized increments indicate candidate actuator locations.

2.3 Model Predictive Path Integral Control (MPPI)

We next summarize the MPPI control framework. MPPI is a sample-based model predictive control (MPC) method grounded in probabilistic inference (Williams et al. (2018)). MPPI considers the following nonlinear dynamical system:

$$\mathbf{x}_{t+1} = M(\mathbf{x}_t, \mathbf{u}_t). \tag{14}$$

At each time step, MPPI optimizes an *open-loop* control sequence over a finite horizon of length T. Let $\mathbf{U} = (\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{T-1})$ denote the *mean* control sequence. MPPI introduces stochastic exploration by sampling control sequences $\mathbf{V} = (\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{T-1})$ from a Gaussian distribution centered at $\mathbf{U}$:

$$q(\mathbf{V} \mid \mathbf{U}, \Sigma) = \prod_{\tau=0}^{T-1} \mathcal{N}(\mathbf{v}_\tau; \mathbf{u}_\tau, \Sigma), \tag{15}$$

where $\Sigma \in \mathbb{R}^{n_u \times n_u}$ is a user-specified positive definite covariance matrix. Equivalently, one may write $\mathbf{v}_\tau = \mathbf{u}_\tau + \boldsymbol{\epsilon}_\tau$ with $\boldsymbol{\epsilon}_\tau \sim \mathcal{N}(\mathbf{0}, \Sigma)$. Given a sampled control sequence $\mathbf{V}$, the system is rolled out according to $\mathbf{x}_{\tau+1} = M(\mathbf{x}_\tau, \mathbf{v}_\tau)$ (starting from the current state), and we define the total trajectory cost functional as

$$S(\mathbf{V}) = \Phi(\mathbf{x}_T) + \sum_{\tau=0}^{T-1} L(\mathbf{x}_\tau, \mathbf{v}_\tau). \tag{16}$$

Under this setup, the stochastic optimal control problem at each time step is

$$\mathbf{U}^* = \arg\min_{\mathbf{U} \in \mathcal{U}} \mathbb{E}_{\mathbf{V} \sim q(\mathbf{V} \mid \mathbf{U}, \Sigma)} \big[S(\mathbf{V}) \big]. \tag{17}$$

MPPI can be derived via a variational free-energy bound. Let $p(\mathbf{V})$ denote a *base* (prior) distribution over control sequences (typically chosen as a Gaussian), and define the free energy

$$F = -\lambda \log \mathbb{E}_{\mathbf{V} \sim p} \left[\exp\left(-\frac{S(\mathbf{V})}{\lambda} \right) \right]. \tag{18}$$

Then, for any distribution $r(\mathbf{V})$, the following inequality holds:

$$F \leq \mathbb{E}_{\mathbf{V} \sim r}[S(\mathbf{V})] + \lambda D_{\mathrm{KL}}(r \| p), \tag{19}$$

where D_{KL} is the Kullback–Leibler divergence. The minimizer of the right-hand side of (19) is the optimal distribution

$$q^*(\mathbf{V}) = \frac{1}{\eta} \exp\left(-\frac{S(\mathbf{V})}{\lambda} \right) p(\mathbf{V}), \tag{20}$$

and substituting $r(\mathbf{V}) = q^*(\mathbf{V})$ into (19) yields equality (see Williams et al. (2018) for details). In practice, MPPI restricts $r(\mathbf{V})$ to a Gaussian family with fixed covariance Σ and optimizes only the mean $\mathbf{U}$. This corresponds to the KL projection

$$\hat{\mathbf{U}} = \arg\min_{\mathbf{U} \in \mathcal{U}} D_{\mathrm{KL}}\big(q^* \| q(\mathbf{V} \mid \mathbf{U}, \Sigma) \big), \tag{21}$$

whose minimizer satisfies, for fixed Σ,

$$\hat{\mathbf{U}} = \mathbb{E}_{\mathbf{V}\sim q^*}[\mathbf{V}]. \tag{22}$$

Since direct sampling from q^* is intractable, we estimate (22) by Monte Carlo sampling with importance weights. When samples $\{\mathbf{V}_k\}_{k=1}^K$ are drawn from the base distribution $p(\mathbf{V})$, (22) can be written as a normalized weighted average:

$$\hat{\mathbf{U}} \approx \sum_{k=1}^{K} \bar{\omega}(\mathbf{V}_k)\,\mathbf{V}_k, \tag{23}$$

with log-weights and normalized weights

$$\omega(\mathbf{V}_k) = -\frac{1}{\lambda}\,S(\mathbf{V}_k), \tag{24}$$

$$\bar{\omega}(\mathbf{V}_k) = \frac{\exp(\omega(\mathbf{V}_k))}{\sum_{j=1}^{K}\exp(\omega(\mathbf{V}_j))}. \tag{25}$$

(If samples are drawn from a proposal distribution $g(\mathbf{V}) \neq p(\mathbf{V})$, the weights are modified by the standard importance-sampling factor $p(\mathbf{V}_k)/g(\mathbf{V}_k)$.)

Typically, only the first element of the optimized control sequence $\hat{\mathbf{U}}$ is applied as the control input at each time step, after which the horizon is shifted forward and the procedure is repeated. As can be seen from (23)–(25), the weights for all samples can be computed in parallel, allowing efficient implementation on GPUs. Moreover, MPPI does not require gradient information of the dynamics or cost function and can be implemented using only forward simulations of the model.

3 EKG-MPPI: Proposed hybrid control method

3.1 Overall framework

EnKC provides a principled framework for determining small yet effective perturbations at each control step. However, EnKC computes the control increment by (approximately) linearizing the system dynamics over a finite prediction horizon, and its performance may therefore deteriorate when the dynamics are strongly nonlinear. To overcome this limitation, we propose a hybrid control scheme, termed *EnKC-guided MPPI* (EKG-MPPI), which combines EnKC with MPPI. In contrast to EnKC, MPPI evaluates candidate perturbations via nonlinear forward rollouts and thus can handle nonlinear dynamics without linearization. Nevertheless, as pointed out by Power and Berenson (2022), MPPI may suffer from poor sampling efficiency when applied without informative prior knowledge of the control distribution. This issue is particularly severe in high-dimensional geophysical systems, where the control space is large and naive sampling may fail to discover effective perturbations under a realistic computational budget.

EKG-MPPI addresses this issue by using the EnKC perturbation as *prior information* for MPPI. Intuitively, EnKC provides a sparse and physically informed guess of (i) *where* the system is most sensitive and (ii) *how strongly* it should be perturbed. We embed this information into the mean and variance (or covariance) of Gaussian sampling distributions, from which MPPI

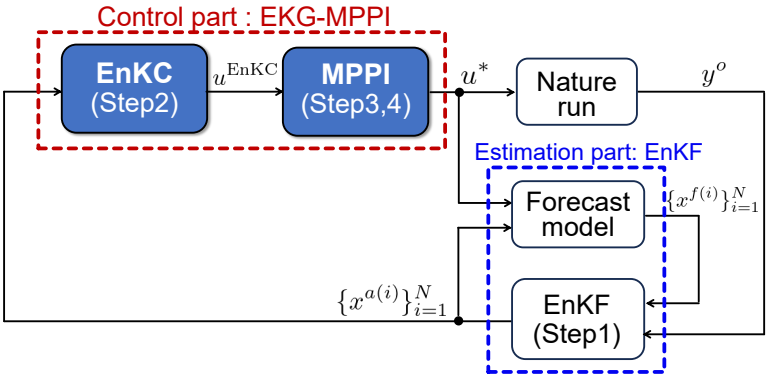


Figure 1. Block diagram of the proposed EKG-MPPI.

draws candidate actuator locations and magnitudes. MPPI then refines the perturbation through sampling-based optimization based on nonlinear rollouts, without any linearization or gradient computation.

The proposed EKG-MPPI scheme is implemented within the CSE framework, and the overall workflow is summarized in Fig. 1. At each data-assimilation time, we first apply the EnKF to assimilate observations and estimate the current state (Step 1). We then run EnKC using the analysis ensemble to obtain a sparse perturbation (Step 2). This perturbation is interpreted as a candidate actuator configuration and mapped to the parameters of Gaussian sampling distributions for actuator location and control magnitude. MPPI samples multiple candidate control inputs from these distributions, evaluates their performance over a prediction horizon, and computes a weighted average to obtain the final control input (Steps 3–4). In the CSE framework, the model trajectory obtained by integrating the forecast model without any control perturbation is referred to as the *nature run*, which serves as a proxy for the true atmosphere. The computed control is applied both to the nature run and to the analysis ensemble, after which the forecast–assimilation–control loop continues to the next time step.

3.2 Algorithmic steps (Step 1–5)

In this subsection we describe the EKG-MPPI procedure at a single control time t. We denote the ensemble size by N and the current analysis ensemble by $\{x_t^{a(i)}\}_{i=1}^N$.

Step 1: EnKF state estimation

As in standard CSE studies, accurate estimation of the system state is a prerequisite for effective control. At each assimilation time t, we apply the EnKF to assimilate the available observations y_t^o and update the forecast ensemble. The EnKF update is

given in Section 2.1 and yields the analysis ensemble $\{x_t^{a(i)}\}_{i=1}^N$. We then compute the analysis ensemble mean

$$\bar{x}_t^a = \frac{1}{N}\sum_{i=1}^{N} x_t^{a(i)}, \tag{26}$$

which serves as the current state estimate used to initialize the subsequent control computation.

Step 2: EnKC-based control perturbation

In the second step, we apply EnKC (Section 2.2) to compute a control perturbation that will serve as prior information for MPPI. EnKC minimizes the quadratic cost function in (8) via an EnKS, interpreting the control objective as a pseudo–observation. The resulting control perturbation $x_t^c - \bar{x}_t^a$ is then sparsified by the adaptive thresholding procedure described in Section 2.2. Among the information contained in the sparsified control vector, the key components for localized intervention are (i) the indices of its nonzero entries, which indicate candidate actuator locations, and (ii) the corresponding magnitudes, which represent nominal control amplitudes at those locations. To simplify the presentation, we set $\Lambda = 1$ in (13), so that only the grid point with the largest perturbation remains nonzero. The resulting sparse control vector is denoted by u_t^{EnKC}, and encodes both the actuator location and its nominal control magnitude to be exploited by MPPI.

Step 3: EnKC-informed sampling of actuator location and magnitude

The third step embeds the information contained in u_t^{EnKC} into the sampling distributions from which MPPI draws candidate control inputs. In this study, we parameterize each candidate actuator configuration by its location and magnitude. Let $\mu_{\mathrm{loc},t}$ and $\Sigma_{\mathrm{loc},t}$ denote the mean and variance (scalar in the single-actuator case) that determine the sampling distribution of actuator locations, and let $\mu_{\mathrm{mag},t}$ and $\Sigma_{\mathrm{mag},t}$ denote the corresponding quantities for the control magnitude. We define

$$\begin{aligned}
\mu_{\mathrm{loc},t} &= f_{\mathrm{loc}}\big(u_t^{\mathrm{EnKC}}\big), \qquad \Sigma_{\mathrm{loc},t} = g_{\mathrm{loc}}\big(u_t^{\mathrm{EnKC}}\big), \\
\mu_{\mathrm{mag},t} &= f_{\mathrm{mag}}\big(u_t^{\mathrm{EnKC}}\big), \qquad \Sigma_{\mathrm{mag},t} = g_{\mathrm{mag}}\big(u_t^{\mathrm{EnKC}}\big),
\end{aligned} \tag{27}$$

where f_{loc} and f_{mag} map the EnKC output to the means of the sampling distributions for the actuator location and magnitude, respectively, and g_{loc} and g_{mag} define the corresponding sampling variances used by MPPI. Using the parameters in (27), we draw K_{MPPI} samples of actuator locations and magnitudes,

$$l_{t,i} \sim \mathcal{G}(l \mid \mu_{\mathrm{loc},t}, \Sigma_{\mathrm{loc},t}), \quad m_{t,i} \sim \mathcal{G}(m \mid \mu_{\mathrm{mag},t}, \Sigma_{\mathrm{mag},t}), \quad i = 1, 2, \ldots, K_{\mathrm{MPPI}}, \tag{28}$$

where $\mathcal{G}(\cdot \mid \mu, \Sigma)$ denotes a Gaussian distribution. Because actuator locations correspond to discrete grid indices, we round $l_{t,i}$ to the nearest integer and clip it to the valid index range; with a slight abuse of notation, we denote the resulting index again by $l_{t,i}$. Each pair $(l_{t,i}, m_{t,i})$ specifies one candidate actuator configuration for MPPI.

Step 4: MPPI rollout and importance weighting

In the fourth step, we evaluate the sampled actuator configurations via MPPI rollouts and compute their importance weights. Let $x_t = \bar{x}_t^a$ denote the current state estimate (i.e., the state from which MPPI rolls out). Let n_u denote the dimension of the

control input, and let $e_\ell \in \mathbb{R}^{n_u}$ be the ℓ-th standard basis vector. For each sample $(l_{t,i}, m_{t,i})$, we construct a sparse control vector

$$u_{t,i} = m_{t,i}\, e_{l_{t,i}}. \tag{29}$$

Starting from $x_{t,0}^{(i)} = x_t$, we propagate the system over the MPPI prediction horizon T_{MPPI} as

$$x_{t,1}^{(i)} = M\big(x_{t,0}^{(i)}, u_{t,i}\big), \tag{30}$$

$$x_{t,\tau+1}^{(i)} = M\big(x_{t,\tau}^{(i)}, \mathbf{0}\big), \quad \tau = 1, 2, \ldots, T_{\mathrm{MPPI}} - 1, \tag{31}$$

where $\mathbf{0}$ denotes the zero control input. That is, the perturbation is applied only at the first rollout step, and no further input is provided during the remaining rollout. This reflects the single-step actuation setting used in our CSEs and also reduces the effective dimension of the control space, thereby improving sampling efficiency.

Let $S(u_{t,i}, x_t, M)$ be a state-dependent cost functional that quantifies the performance of the i-th control sample. In our implementation, the EnKC-informed Gaussian distribution in Step 3 is used as the base distribution of MPPI, and therefore no importance-sampling correction term is required. The (unnormalized) log-weight is computed as

$$\omega(u_{t,i}) = -\frac{1}{\lambda}\, S(u_{t,i}, x_t, M)\,, \tag{32}$$

where λ is the temperature parameter in MPPI. The normalized weight is then computed as

$$\bar{w}_{t,i} = \frac{\exp\big(\omega(u_{t,i})\big)}{\sum_{k=1}^{K_{\mathrm{MPPI}}} \exp\big(\omega(u_{t,k})\big)}. \tag{33}$$

Finally, we first compute the weighted-average control vector

$$\tilde{u}_t = \sum_{i=1}^{K_{\mathrm{MPPI}}} \bar{w}_{t,i}\, u_{t,i}, \tag{34}$$

and then project it to a single-actuator (1-sparse) control by keeping only the largest-magnitude component:

$$l_t^* = \arg\max_{\ell} |\tilde{u}_t(\ell)|\,, \tag{35}$$

$$u_t^* = \tilde{u}_t(l_t^*)\, e_{l_t^*}. \tag{36}$$

Step 5: Feedback

In the final step, the optimal control u_t^* computed in (36) is applied both to the nature run and to each member of the analysis ensemble, i.e.

$$x_t^{a(i)} \leftarrow x_t^{a(i)} + u_t^*, \quad i = 1, 2, \ldots, N. \tag{37}$$

The updated ensemble is then advanced by the forecast model until the next assimilation time, at which point the procedure returns to Step 1.

Algorithm 1 EKG-MPPI algorithm

Require: initial analysis ensemble $\{\mathbf{x}_0^{a(i)}\}_{i=1}^N$; localization scale L; observation-error covariance matrix $\mathbf{R}$; forecast model M; observation operator H; operator that maps the state variables to the control criteria H_c; control-weight matrix $\mathbf{R}_c$; EnKC prediction horizon T_{EnKC}; MPPI prediction horizon T_{MPPI}; sparsity coefficient Λ; state-cost functional S; embedding functions $f_{\mathrm{loc}}, g_{\mathrm{loc}}, f_{\mathrm{mag}}, g_{\mathrm{mag}}$; number of MPPI samples K_{MPPI}; temperature parameter λ; total number of assimilation steps T_{sim}.

1: **for** $t = 1, 2, \ldots, T_{\mathrm{sim}}$ **do**
2: Obtain observation $\mathbf{y}_t^o$.
3: $\{\mathbf{x}_t^{a(i)}\}_{i=1}^N \leftarrow \mathbf{EnKF}\big(\{\mathbf{x}_{t-1}^{a(i)}\}_{i=1}^N, \mathbf{y}_t^o, L, \mathbf{R}, M, H\big)$.
4: Compute the analysis ensemble mean $\bar{\mathbf{x}}_t^a \leftarrow \frac{1}{N}\sum_{i=1}^N \mathbf{x}_t^{a(i)}$.
5: $u_t^{\mathrm{EnKC}} \leftarrow \mathbf{EnKC}\big(\{\mathbf{x}_t^{a(i)}\}_{i=1}^N, T_{\mathrm{EnKC}}, \Lambda, \mathbf{R}_c, M, H_c\big)$.
6: Compute $\mu_{\mathrm{loc},t}$, $\Sigma_{\mathrm{loc},t}$, $\mu_{\mathrm{mag},t}$, $\Sigma_{\mathrm{mag},t}$ according to (27).
7: **for** $i = 1, 2, \ldots, K_{\mathrm{MPPI}}$ **do**
8: Sample $l_{t,i} \sim \mathcal{G}(l \mid \mu_{\mathrm{loc},t}, \Sigma_{\mathrm{loc},t})$, $m_{t,i} \sim \mathcal{G}(m \mid \mu_{\mathrm{mag},t}, \Sigma_{\mathrm{mag},t})$.
9: Construct a sparse control vector $u_{t,i} \leftarrow m_{t,i}\, e_{l_{t,i}}$, where e_ℓ denotes the ℓ-th standard basis vector.
10: Evaluate the cost $S(u_{t,i}, \bar{x}_t^a, M)$ and compute the (unnormalized) log-weight $\omega(u_{t,i})$ using (32).
11: **end for**
12: Normalize the weights $\bar{w}_{t,i}$ using (33).
13: Compute the weighted-average control vector $\tilde{u}_t \leftarrow \sum_{i=1}^{K_{\mathrm{MPPI}}} \bar{w}_{t,i}\, u_{t,i}$.
14: Select the actuator location $l_t^* \leftarrow \arg\max_\ell |\tilde{u}_t(\ell)|$.
15: Set the single-actuator control $u_t^* \leftarrow \tilde{u}_t(l_t^*)\, e_{l_t^*}$.
16: Apply u_t^* to the nature run and update each analysis ensemble member via $x_t^{a(i)} \leftarrow x_t^{a(i)} + u_t^*, \;\; i = 1, \ldots, N$.
17: Advance the forecast model M to the next assimilation time.
18: **end for**

3.3 Pseudocode of EKG-MPPI

The EKG-MPPI procedure described above can be summarized by Algorithm 1, where **EnKF** (line 3) denotes the ensemble Kalman filter used for state estimation and **EnKC** (line 5) denotes the ensemble Kalman control scheme that computes sparse optimal perturbations as described in Section 2.2.

3.4 Single-step setting and motivation for MPPI

In the current implementation, the control perturbation is applied only at the first rollout step, and no further input is provided during the remaining horizon. The resulting optimization problem at each control decision is therefore effectively single-step. This approach is adopted as a proof of concept for localized intervention in chaotic flows and also reduces the effective dimension of the control space, thereby improving sampling efficiency. Nevertheless, we retain the MPPI formulation because

rollout evaluation and importance-weight computation can be performed in parallel for a batch of samples, which is advantageous when forward simulation is the dominant computational cost. In addition, the same algorithmic framework extends naturally to multi-step control sequences and multiple actuators. To examine whether this choice remains competitive even in the present single-step setting, we provide a direct comparison with a standalone Bayesian optimization (BO) baseline (for details, see Section 4.1.2 and appendix A).

4 Numerical Experiments

We demonstrate the effectiveness of EKG-MPPI through numerical experiments using the Lorenz–96 model and the SQG model.

4.1 Lorenz–96 model

Following the methodology of Sun et al. (2023); Sawada (2024a), we conduct a control simulation experiment (CSE) aimed at mitigating extreme values in the Lorenz–96 system (Lorenz (1995)). The Lorenz–96 system is governed by

$$\frac{dX_k}{dt} = (X_{k+1} - X_{k-2})X_{k-1} - X_k + F, \quad k \in \{1, 2, \ldots, K\}, \tag{38}$$

where $K = 40$ in this study, and the external forcing parameter is set to $F = 8.0$. Cyclic boundary conditions are imposed such that the indices are taken modulo K, i.e. $X_{k+K} = X_k$. Equation (38) is numerically integrated using the fourth-order Runge–Kutta method with a time step of $\Delta t = 0.05$. We perform numerical simulations over 160,600 model time steps. During the first 14,600 time steps, neither EnKC nor EKG-MPPI is applied; only EnKF is used to synchronize the estimated state with the uncontrolled nature run (spin-up period). Control is activated for the remaining 146,000 time steps, during which EnKF, EnKC, and EKG-MPPI are all executed. All diagnostics reported below are computed over this controlled period.

4.1.1 EnKC vs. EKG-MPPI

In this subsection, we evaluate the effectiveness of the proposed EKG-MPPI method in comparison with EnKC. The hyperparameter settings for the two methods are as follows. For EnKC, observations are assumed to be available at every other grid point. The observation-error covariance matrix $\mathbf{R}$ is set to the identity matrix. The ensemble size is $N = 40$, and the localization length scale is $L = 2$. The prediction horizon is set to $T_{\mathrm{EnKC}} = 0.2$, corresponding to four time steps. The state threshold is set to $r_{t+T_{\mathrm{EnKC}}} = 12$. The pseudo-observation matrix is defined so that grid points whose predicted states at time $t + T_{\mathrm{EnKC}}$ exceed the threshold are selected as observation points. The control-weight matrix $\mathbf{R}_c$ is set to 10^{-4}.

For EKG-MPPI, the EnKC-related parameters are set identically to those of EnKC. In addition, the temperature parameter is set to $\lambda = 0.25$, the number of samples is $K_{\mathrm{MPPI}} = 40$, and the MPPI prediction horizon is set to $T_{\mathrm{MPPI}} = 4$. The embedding functions in (27) are specified as

$$\begin{aligned} f_{\mathrm{loc}}\big(u_t^{\mathrm{EnKC}}\big) &= \mathrm{argnz}\big(u_t^{\mathrm{EnKC}}\big), & g_{\mathrm{loc}}\big(u_t^{\mathrm{EnKC}}\big) &= 5.0, \\ f_{\mathrm{mag}}\big(u_t^{\mathrm{EnKC}}\big) &= u_t^{\mathrm{EnKC}}(f_{\mathrm{loc}}\big(u_t^{\mathrm{EnKC}}\big)), & g_{\mathrm{mag}}\big(u_t^{\mathrm{EnKC}}\big) &= 0.1, \end{aligned} \tag{39}$$

where $\mathrm{argnz}(\cdot)$ returns the index of the nonzero entry of its argument. For the initial solution of MPPI, we adopt the perturbations obtained by EnKC, as represented by f_{loc} and f_{mag}. The running state-cost function is defined as

$$S(\mathbf{x}_t) = \|\max(\mathbf{x}_t - 12, 0)\|_1 . \tag{40}$$

Under these settings, we conducted 10 simulations with different random seeds. For each method, we computed the mean and standard deviation over the 10 runs of (i) the number of state components exceeding the threshold value of 12 and (ii) the control-input magnitude. The results are summarized in Table 1. EKG-MPPI achieved a lower extreme-event count (7622 ± 146) than EnKC (7820 ± 142), while also requiring a smaller control-input magnitude (0.166 ± 0.002 for EKG-MPPI versus 0.191 ± 0.001 for EnKC).

Table 1. Comparison of EKG-MPPI and EnKC in the Lorenz–96 control simulation experiment. The second column reports the extreme-event count, defined as $\#\{(t,k) : X_{t,k} \geq 12\}$, where $\#\{\cdot\}$ denotes set cardinality. The third column reports the input magnitude, defined as $\frac{1}{N_{\mathrm{ctrl}}} \sum_{t \in \mathcal{T}_{\mathrm{ctrl}}} \sum_{k} |u_{t,k}|$, where $\mathcal{T}_{\mathrm{ctrl}}$ denotes the set of time steps at which control is applied during the simulation horizon T_{sim}, and $N_{\mathrm{ctrl}} = |\mathcal{T}_{\mathrm{ctrl}}|$. All values are shown as mean $\pm$ standard deviation over 10 simulations with different random seeds.

Method	Extreme-event count ($X \geq 12$)	Input magnitude
EKG-MPPI	7622 ± 146	0.166 ± 0.002
EnKC	7820 ± 142	0.191 ± 0.001

These results demonstrate that EKG-MPPI outperforms EnKC in terms of both suppressing extreme events and reducing control effort. We attribute the improvement over EnKC to the nonlinear evaluation introduced in the MPPI refinement stage.

4.1.2 Comparison with Bayesian optimization in the single-step setting

Since the control perturbation is applied only at the first rollout step, the optimization performed at each control decision reduces to the selection of a single localized perturbation. Bayesian optimization (BO) is a standard framework for sample-efficient optimization of expensive black-box functions (see, e.g., Shahriari et al. (2016)), and thus it provides a natural baseline for the present single-step setting. We therefore compare EKG-MPPI with a standalone BO controller to assess whether the MPPI-based formulation remains advantageous even in this simplified setting. For a fair comparison, the BO baseline optimizes the same single-step control parameterization as EKG-MPPI, using the same prediction horizon and the same state-cost function. The objective function used by the BO baseline is designed to balance the severity of extreme events against the intervention cost. Specifically, the cost function is defined as

$$c(x_t, u_t) = \sum_{i=1}^{T_{\mathrm{BO}}} \sum_{j=1}^{K} \max\left(x_{t+i}^{(j)} - r_{\mathrm{th}}, 0\right) + \alpha \|u_t\|_1, \tag{41}$$

where $x_{t+i}^{(j)}$ denotes the j-th component of the predicted state, K is the state dimension, and T_{BO} is the prediction horizon. The state trajectory is obtained by propagating the model dynamics as

$$\begin{aligned} x_{t+1} &= M(x_t + u_t), \\ x_{t+i} &= M(x_{t+i-1}), \qquad i = 2, \ldots, T_{\mathrm{BO}}. \end{aligned} \tag{42}$$

The first term measures the cumulative exceedance above the threshold over the prediction horizon, while the second term penalizes the intervention magnitude. The parameter α controls the trade-off between intervention effectiveness and intervention cost. We employ the Expected Improvement (EI) acquisition function Jones et al. (1998), set the search interval to $[-1, 1]$, use $\alpha = 1$, and perform 7 BO iterations at each control decision. Although this is a modest BO budget, it already leads to a substantially larger computation time per decision than that of EKG-MPPI; see Table 3 for the runtime comparison.

Table 2 summarizes the results. As in Section 4.1.1, the experiments are repeated over 10 different random seeds. EKG-MPPI yields fewer extreme events than BO (7622 ± 146 vs. 8299 ± 255) and a smaller mean control-input magnitude (0.166 ± 0.002 vs. 0.340 ± 0.004). These results indicate that EKG-MPPI remains effective in the present single-step setting, outperforming the standalone BO baseline in both reported metrics. For completeness, Appendix A reports the corresponding comparison between standalone BO and vanilla MPPI under the same single-step setting.

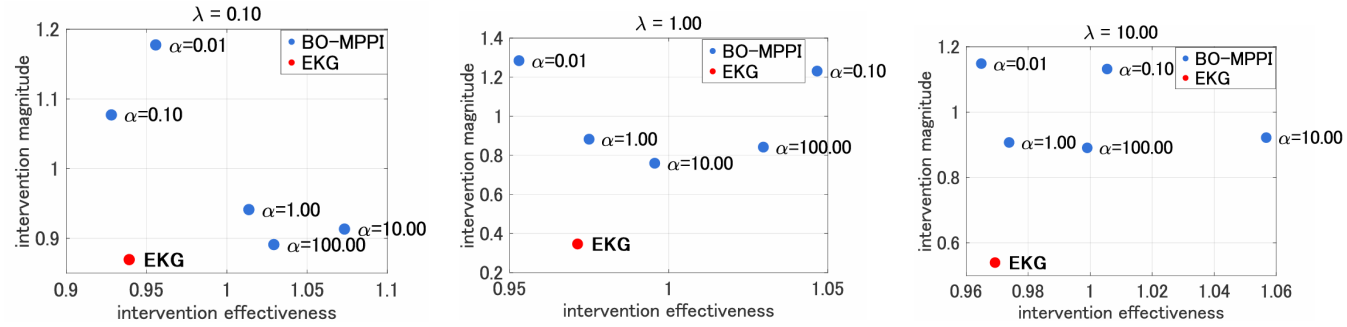
Table 2. Comparison between EKG-MPPI and a standalone BO baseline in the single-step Lorenz–96 control setting. The second and third columns report the extreme-event count and the mean input magnitude, respectively, defined in the same manner as in Table 1.

Method	Extreme-event count ($X \geq 12$)	Mean input magnitude
EKG-MPPI	7622 ± 146	0.166 ± 0.002
BO	8299 ± 255	0.340 ± 0.004

4.1.3 BO-informed prior vs. EnKC-informed prior in MPPI

To further examine the prior design used in EKG-MPPI, we introduce an additional baseline, denoted BO-MPPI, in which Bayesian optimization is used to estimate the intervention location and magnitude that define the prior supplied to MPPI. Unlike the standalone BO baseline in Section 4.1.2, the purpose of this comparison is not to contrast BO with EKG-MPPI. Rather, because BO-MPPI and EKG-MPPI share the same MPPI refinement stage, this comparison isolates the effect of the prior used to guide MPPI. BO-MPPI therefore serves as an ablation baseline for the prior construction.

For the BO component of BO-MPPI, the optimization iterations and the search interval for the intervention are set in the same way as in Section 4.1.2. The hyperparameters of EKG-MPPI are set as in Section 4.1.1. Under these settings, for each $\lambda \in \{0.1, 1.0, 10\}$, we conduct experiments by varying the BO-MPPI parameter α over $\{0.01, 0.1, 1, 10, 100\}$. For ease of visualization, both the average intervention magnitude and the intervention-effectiveness metric are rescaled; see Appendix B for details.



(a) $\lambda = 0.1$ (b) $\lambda = 1.0$ (c) $\lambda = 10$

Figure 2. Comparison between BO-MPPI and EKG-MPPI for different values of λ. From left to right, $\lambda = 0.1$, 1.0, and 10.

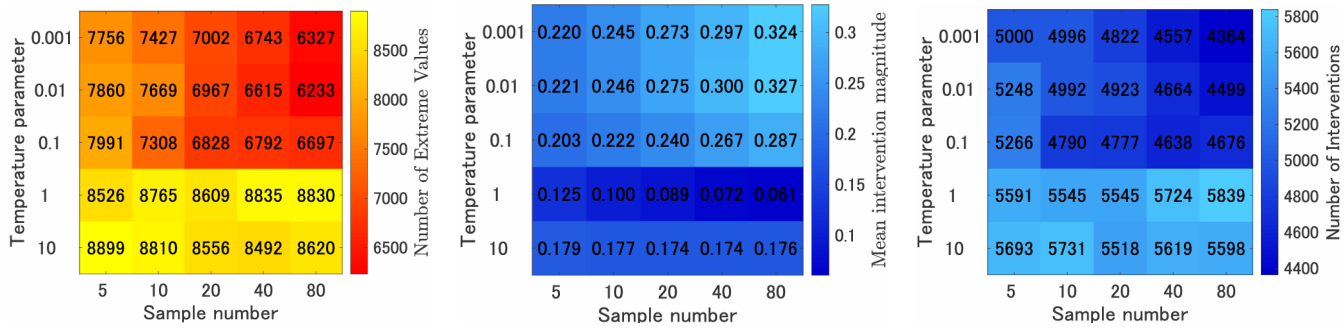
Figure 2 compares EKG-MPPI and BO-MPPI for $\lambda \in \{0.1, 1.0, 10\}$. In each panel, the EKG-MPPI result is located below the empirical lower envelope, i.e., the Pareto-like trade-off curve, formed by the BO-MPPI results obtained by varying α. This indicates that, for a comparable level of intervention effectiveness, EKG-MPPI achieves a smaller average intervention magnitude than BO-MPPI. In other words, the EnKC-informed prior yields a more favorable trade-off between intervention effectiveness and control magnitude than the BO-informed prior across the tested values of λ. These results suggest that the advantage of EKG-MPPI is robust to the choice of λ rather than being driven by a particular hyperparameter setting.

4.1.4 Sensitivity analysis of EKG-MPPI hyperparameters

We next examine the sensitivity of EKG-MPPI to its two main hyperparameters: the temperature parameter λ and the number of MPPI samples K_{MPPI}. We vary λ over $\{10, 1, 0.1, 0.01, 0.001\}$ and K_{MPPI} over $\{5, 10, 20, 40, 80\}$. For each configuration, we record the number of threshold exceedances, the mean control-input magnitude, and the number of time steps at which control is applied. The results are shown in Figure 3.

Figure 3 shows a clear overall tendency: decreasing λ increases the mean control-input magnitude while reducing the frequency of control application. In MPPI, λ determines how strongly the importance weights concentrate on low-cost samples. Smaller values of λ therefore produce more peaked weights and drive the update toward more aggressive perturbations identified in the sampled rollouts. In EKG-MPPI, the sampling distribution is centered on the EnKC-informed proposal u_t^{EnKC}. A smaller λ thus induces a stronger shift away from this EnKC-guided proposal toward samples with particularly low rollout cost. This tendency becomes more pronounced when λ is very small (e.g., 0.01 or 0.001). In that regime, increasing K_{MPPI} tends to further increase the mean control-input magnitude and decrease the control frequency. A plausible explanation is that a larger sample set more often contains rare but highly effective perturbations; when λ is small, the weight update becomes strongly concentrated on such samples, thereby amplifying the resulting control magnitude. Figure 3 also indicates that smaller λ and larger K_{MPPI} generally reduce the number of threshold exceedances, suggesting stronger suppression of extremes. At the same time, the mean input magnitude exhibits a non-monotonic dependence on λ, with the smallest value occurring around $\lambda = 1$ in

the present experiment. Although one might expect a larger value such as $\lambda = 10$ to yield smaller inputs, an excessively large λ produces diffuse weights and hence overly conservative updates. Such conservative control may allow extremes to persist, which can in turn require larger or more sustained interventions later.



(a) Number of Extreme Values (b) Average Input (c) Frequency Input

Figure 3. Hyperparameter configuration map displaying the effects of varying λ and sample size on control outcomes.

4.1.5 Computation time

We next compare the computation time of EnKC, BO, BO-MPPI, and EKG-MPPI. For EnKC, BO and EKG-MPPI, the parameter settings are the same as those in Sect. 4.1.1 and in Sect. 4.1.2. For BO-MPPI, the search interval for the intervention is set to $[-1.0, 1.0]$, and the number of BO iterations is set to 7. At each control decision, we measure the time required to compute the control input and summarize it by its mean and standard deviation. The results are listed in Table 3. For EKG-MPPI, the mean runtime per control decision is $(6.30 \pm 5.23) \times 10^{-5}$ s for the EnKC part and $(3.24 \pm 1.17) \times 10^{-4}$ s for the MPPI rollout and weight computation, giving a total mean runtime of 3.87×10^{-4} s per decision. Thus, approximately 16% of the runtime is attributable to the EnKC part and 84% to the MPPI part. This indicates that the dominant computational cost in EKG-MPPI arises from rollout evaluation rather than from construction of the EnKC-informed prior. The additional overhead introduced by the prior estimation is therefore modest in the present setting. Table 3 also shows that EKG-MPPI is substantially faster than BO-MPPI. The mean runtime of BO-MPPI is 0.216 ± 0.104 s per control decision, whereas the total mean runtime of EKG-MPPI is 3.87×10^{-4} s, making BO-MPPI approximately 5.6×10^{2} times slower in the present experiment. Moreover, the total runtime of EKG-MPPI is much shorter than one model time step, $\Delta t = 0.05$, indicating that the method is computationally feasible for the present online-control setting. Finally, the present experiment uses $K_{\mathrm{MPPI}} = 40$ samples. Under ideal parallel execution, the MPPI component could in principle be reduced to approximately one-fortieth of its current runtime, i.e., about 8.1×10^{-6} s. This observation further suggests that EKG-MPPI is well suited to parallel hardware when forward rollout evaluation dominates the computational cost.

Table 3. Mean computation time per control decision (mean ± standard deviation). For EKG-MPPI, the runtime is decomposed into the EnKC-based prior construction part and the MPPI rollout/weight computation part.

Method	EnKC	BO	BO-MPPI	EKG-MPPI	
				EnKC part	MPPI part
Time [s]	$(3.03 \pm 7.61) \times 10^{-5}$	$(6.64 \pm 0.82) \times 10^{-2}$	$(1.72 \pm 0.92) \times 10^{-1}$	$(6.30 \pm 5.23) \times 10^{-5}$	$(3.24 \pm 1.17) \times 10^{-4}$

4.2 Surface quasi-geostrophic model

The quasi-geostrophic (QG) model is a standard framework for describing mesoscale barotropic and baroclinic dynamics; for a comprehensive review, see Vallis (2017). The surface quasi-geostrophic (SQG) model is derived from the QG model under the assumption of uniform interior potential vorticity. A key feature of the SQG model is that the surface buoyancy acts as an active tracer from which the horizontal velocity field is diagnosed. All numerical experiments follow the setup and parameter values in Resseguier et al. (2017).

4.2.1 SQG dynamics and diagnostic velocity

Let $b(x,t)$ denote the surface buoyancy on a doubly periodic domain $x = (x,y) \in \mathbb{T}^2$. In the SQG model, b evolves as an active tracer and the velocity is diagnosed from b via a streamfunction ψ:

$$\partial_t b + v \cdot \nabla b = \mathcal{D}(b) + f, \tag{43}$$

$$v = \nabla^\perp \psi = (-\partial_y \psi,\, \partial_x \psi)\,, \tag{44}$$

$$b = (-\Delta)^{1/2} \psi. \tag{45}$$

Here $\mathcal{D}(b)$ denotes a dissipation operator and f is an external forcing term. Equivalently, in Fourier space, $\widehat{\psi}(k) = |k|^{-1}\widehat{b}(k)$ for $k \neq 0$, with the mean mode set to zero; see Resseguier et al. (2017) for numerical implementation details. We discretize b on a 128×128 grid and denote the vectorized buoyancy field at discrete time t by $x_t \in \mathbb{R}^n$ with $n = 128^2$. Let $M_0 : \mathbb{R}^n \to \mathbb{R}^n$ be the one-step forecast model corresponding to the numerical time integrator of (43) (including the SQG inversion (45)):

$$x_{t+1} = M_0(x_t). \tag{46}$$

It is assumed that the surface buoyancy can be controlled via a small, spatially localized increment. Accordingly, we model the actuation as an additive perturbation applied at the control time t:

$$x_t^+ = x_t + u_t, \tag{47}$$

followed by the uncontrolled model integration. Thus, the controlled forecast model M used in Step 4 (Section 3.2) is defined by

$$M(x_t, u_t) = M_0(x_t + u_t), \qquad M(x_t, 0) = M_0(x_t). \tag{48}$$

This is consistent with the single-step actuation setting assumed in Step 4 (Section 3.2), where the perturbation is applied only at the first rollout step and no further input is provided during the remaining rollout. We consider a single localized actuator at each control time. Let $l_t = (i_t, j_t)$ denote the actuator location on the 128×128 grid and $m_t \in \mathbb{R}$ its magnitude. Let $e_{l_t} \in \mathbb{R}^n$ be the standard basis vector corresponding to grid point l_t (i.e., a Kronecker delta on the vectorized grid). The control input is then

$$u_t = m_t \, e_{l_t}. \tag{49}$$

In the EKG-MPPI/MPPI sampling step, l_t is sampled in $\mathbb{R}^2$ and then rounded to the nearest integer grid index and clipped to the valid range in each coordinate, following the same discretization convention as in Section 3.2. Given a buoyancy state x_t, we diagnose the streamfunction and velocity using (44)–(45), and define the wind-speed magnitude field as

$$w(x_t)(x) = \|v(x_t)(x)\|_2. \tag{50}$$

Let Ω_{tar} denote a prescribed target region (a set of grid points). Our control objective is to suppress wind speed within Ω_{tar}, and we define the running state-cost function as

$$S(x_t) = \mathrm{target}(w(x_t)), \tag{51}$$

where $\mathrm{target}(\cdot)$ aggregates wind speed within Ω_{tar} (e.g., regional mean or maximum). See Resseguier et al. (2017) for numerical details of the SQG inversion and the diagnostic computation of $w(\cdot)$.

4.2.2 Simulation Results

To compare the control performance of EKG-MPPI and EnKC, we conduct control simulations on a 128×128 grid map under eight different target regions: $[85,95] \times [85,95]$, $[85,95] \times [90,100]$, $[85,95] \times [95,105]$, $[90,100] \times [85,95]$, $[90,100] \times [90,100]$, $[90,100] \times [95,105]$, $[95,105] \times [85,95]$ and $[95,105] \times [90,100]$. Hereafter, these experiments are referred to as exp1 through exp8. For each scenario, five simulations are conducted using different random seeds, resulting in a total of 40 simulations. The mean and standard deviation of the maximum wind speed and the average intervention amount are then computed for comparison. The total number of simulation steps in each run is set to 10118, which corresponds to approximately 10 days of real time. For EnKC, the ensemble size is set to $N = 40$, the prediction horizon is set to $T_{\mathrm{EnKC}} = 500$ steps (approximately 12 hours), and the control weight is set to 10. For EKG-MPPI, the number of samples is set to $K_{\mathrm{MPPI}} = 40$, and the prediction horizon is also set to 500 steps. The embedding function into the prior distribution is defined as follows:

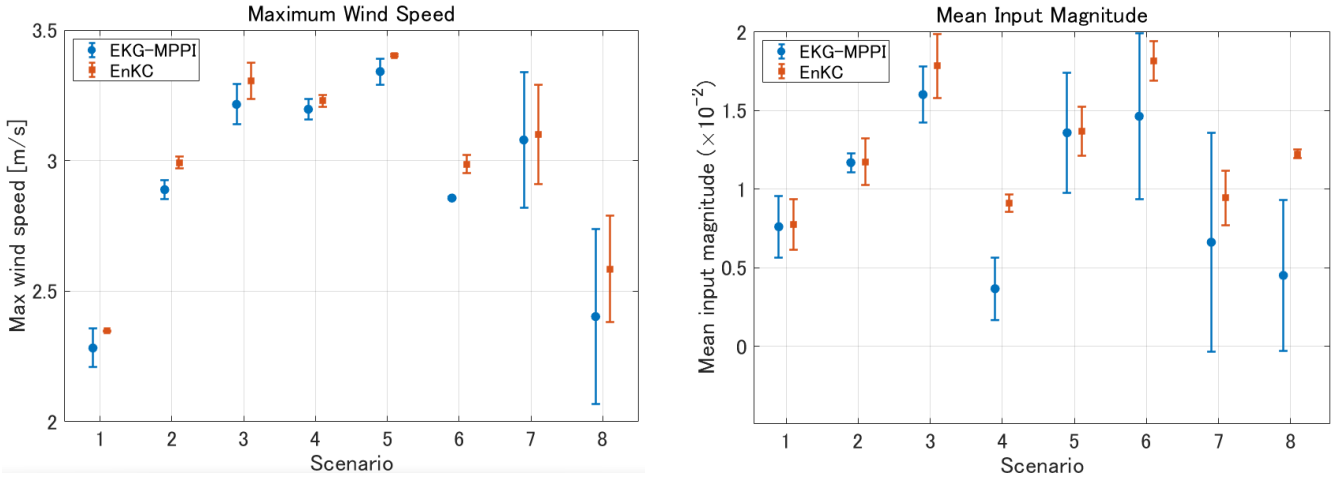
$$\begin{aligned} f_{\mathrm{loc}}\big(u_t^{\mathrm{EnKC}}\big) &= \mathrm{argnz}\big(u_t^{\mathrm{EnKC}}\big), & g_{\mathrm{loc}}\big(u_t^{\mathrm{EnKC}}\big) &= 10.0, \\ f_{\mathrm{mag}}\big(u_t^{\mathrm{EnKC}}\big) &= u_t^{\mathrm{EnKC}}(f_{\mathrm{loc}}\big(u_t^{\mathrm{EnKC}}\big)), & g_{\mathrm{mag}}\big(u_t^{\mathrm{EnKC}}\big) &= \|u_t^{\mathrm{EnKC}}\|_1^2, \end{aligned} \tag{52}$$

The state cost function for EKG-MPPI is defined as follows:

$$S(x_t) = \max(\mathrm{target}(w(x_t)) - w_{th}, 0) \tag{53}$$

where $w(\cdot)$ is a function that converts the buoyancy state x_t into the corresponding wind speed, and $\mathrm{target}(\cdot)$ is a function that computes the magnitude of the wind speed within the target region. Here w_{th} denotes the wind speed threshold, and in this study we set $w_{th} = 2$. For details of the function $w(\cdot)$, the reader is referred to Resseguier et al. (2017).

The results are shown in Figure 4. Focusing on the mean maximum wind speed and the mean intervention magnitude, we observe that EKG-MPPI achieves effective wind-speed suppression with smaller interventions in almost all scenarios. In particular, EKG-MPPI attains lower maximum wind speeds in the target region than EnKC (Figures 4a), while requiring smaller mean control magnitudes (Figures 4b). One reason why EKG-MPPI demonstrates superior control performance compared to EnKC is that EKG-MPPI leverages prior information provided by EnKC to perform MPPI with high sample efficiency, enabling the computation of control inputs that explicitly account for the nonlinear dynamics of the atmospheric system. On the other hand, the standard deviation of the intervention magnitude tends to be larger for EKG-MPPI. A possible reason is the additional exploration introduced by the sampling process in the MPPI component. In large-scale systems such as atmospheric dynamics, control performance is likely to vary significantly depending on the quality of the sampled intervention sequences, which can lead to larger variability in the required control effort. Another interesting observation is that the standard deviation of the maximum wind speed is particularly large in Scenarios 7 and 8. A possible reason is that, in these scenarios, it takes a relatively long time from the beginning of the simulation for the wind speed to intensify. As a result, the effects of interventions at each time step may accumulate over time, leading to more complex behavior of the atmospheric field.



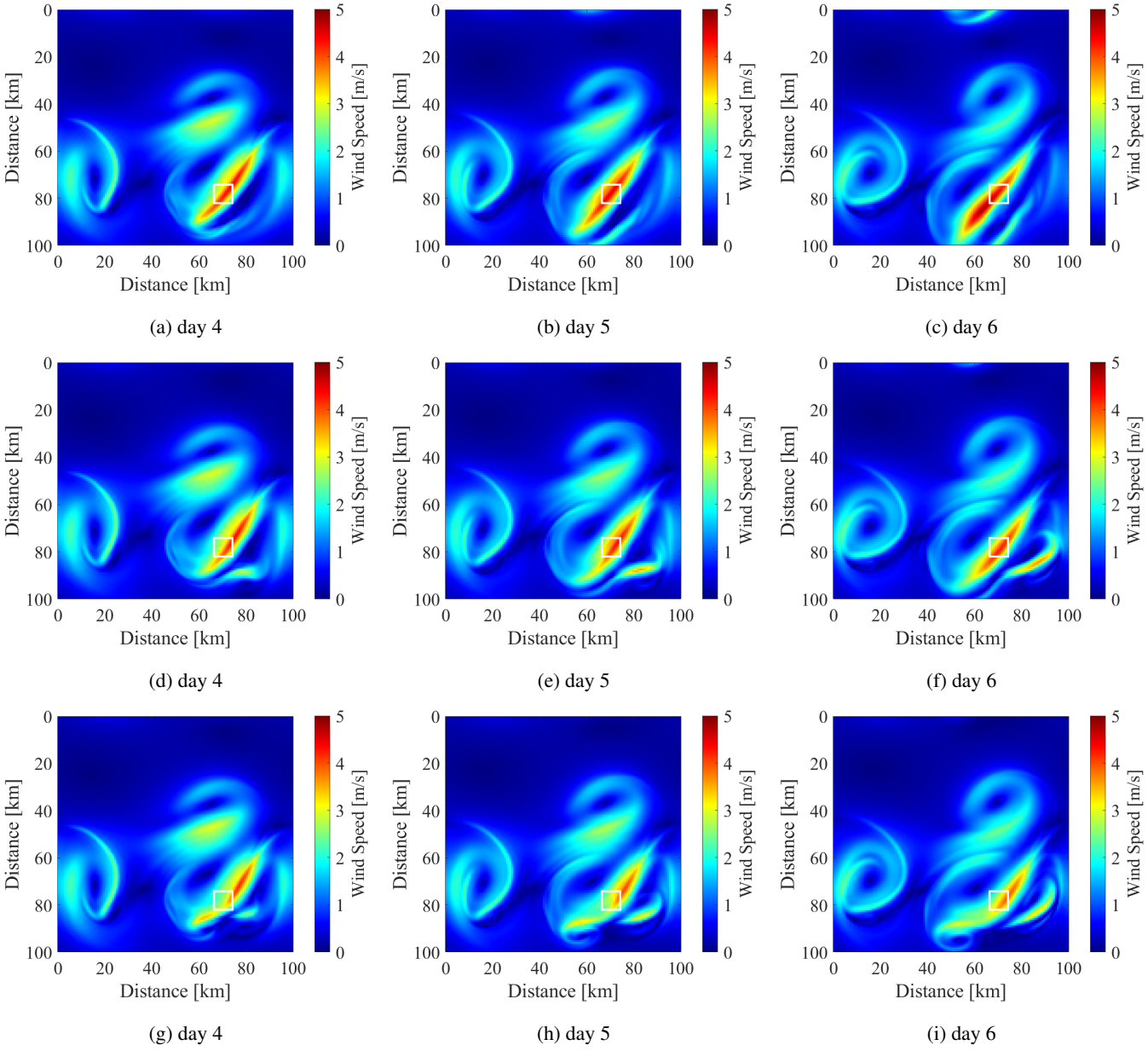
(a) Maximum Wind Speed

(b) Mean Input Magnitude

Figure 4. Mean and standard deviation of the maximum wind speed and the average intervention magnitude for each scenario. Smaller values indicate better performance in terms of extreme-event suppression and control effort, respectively. Note that the figure displays raw values for each scenario rather than percentage differences.

For clarity, Figure 5 shows visualizations of the no-control, EnKC, and EKG-MPPI simulations for the target region $[85, 95] \times [95, 105]$. From Figures 5a, 5b and 5c, it can be observed that strong winds occur in the target region in the absence of control.

From 5d, 5e and 5f, EnKC suppresses the wind speed in the target region compared to the no-control case; however, localized regions of strong wind still remain. From 5g, 5h and 5i, EKG-MPPI achieves stronger wind-speed suppression in the target region than EnKC.



(a) day 4 (b) day 5 (c) day 6

(d) day 4 (e) day 5 (f) day 6

(g) day 4 (h) day 5 (i) day 6

Figure 5. The first, second, and third rows show the time evolution of the wind-speed field for the no-control, EnKC, and EKG-MPPI cases, respectively. The white rectangle denotes the target region for wind-speed suppression.

4.3 Conclusions

We developed EnKC-guided MPPI (EKG-MPPI), a hybrid control scheme that uses EnKC to obtain a sparse candidate perturbation and then refines it by MPPI using nonlinear forward rollouts. The EnKC output is used to shape the sampling distribution in MPPI, so that exploration is concentrated around plausible actuator locations and magnitudes instead of relying on uninformed sampling.

In the Lorenz–96 control simulation experiment, EKG-MPPI reduced the extreme-event count relative to EnKC while requiring a smaller input magnitude. Specifically, for $X \geq 12$, the extreme-event count decreased from 7820 ± 142 for EnKC to 7622 ± 146 for EKG-MPPI, while the input magnitude decreased from 0.191 ± 0.001 to 0.166 ± 0.002. In the SQG experiments over eight target regions, EKG-MPPI also achieved lower maximum wind speeds in the target region than EnKC, while using smaller mean control magnitudes.

The current implementation assumes localized actuation (a single or very sparse actuator) and applies the perturbation as a one-step input within each MPPI rollout, and it does not yet impose hard physical constraints. Future work will address multi-actuator and multi-step actuation, constraint-aware formulations, and evaluation metrics that explicitly quantify non-target impacts and robustness to model/observation uncertainty, in addition to improving computational efficiency for higher-dimensional models.

References

Cotton, W., Zhang, H., Mcfarquhar, G., and Saleeby, S.: Should we consider polluting hurricanes to reduce their intensity, J. Weather Mod., 39, 2007.

Henderson, J. M., Hoffman, R. N., Leidner, S. M., Nehrkorn, T., and Grassotti, C.: A 4D-Var study on the potential of weather control and exigent weather forecasting, Quarterly Journal of the Royal Meteorological Society, 131, 3037–3051, https://doi.org/10.1256/qj.05.72, 2005.

Houtekamer, P. L. and Zhang, F.: Review of the Ensemble Kalman Filter for Atmospheric Data Assimilation, Monthly Weather Review, 144, 4489 – 4532, https://doi.org/10.1175/MWR-D-15-0440.1, 2016.

Jacobson, M. and Kempton, W.: Taming hurricanes with arrays of offshore wind turbines, Nature Climate Change, 4, https://doi.org/10.1038/nclimate2120, 2014.

Jones, D. R., Schonlau, M., and Welch, W. J.: Efficient Global Optimization of Expensive Black-Box Functions, Journal of Global Optimization, 13, 455–492, 1998.

Kawasaki, F. and Kotsuki, S.: Leading the Lorenz 63 system toward the prescribed regime by model predictive control coupled with data assimilation, Nonlinear Processes in Geophysics, 31, 319–333, https://doi.org/10.5194/npg-31-319-2024, 2024.

Latham, J., Parkes, B., Gadian, A., and Salter, S.: Weakening of hurricanes via marine cloud brightening (MCB), Atmospheric Science Letters, 13, 231–237, https://doi.org/10.1002/asl.402, 2012.

Lorenz, E.: Predictability: a problem partly solved, Ph.D. thesis, Shinfield Park, Reading, 1995.

Miller, J., Tang, A., Tran, T. L., Prinsley, R., and Howden, M.: The Feasibility and Governance of Cyclone Interventions, Climate Risk Management, 41, 100 535, https://doi.org/https://doi.org/10.1016/j.crm.2023.100535, 2023.

Miyoshi, T. and Sun, Q.: Control simulation experiment with Lorenz's butterfly attractor, Nonlinear Processes in Geophysics, 29, 133–139, https://doi.org/10.5194/npg-29-133-2022, 2022.

Ouyang, M., Tokuda, K., and Kotsuki, S.: Reducing manipulations in a control simulation experiment based on instability vectors with the Lorenz-63 model, Nonlinear Processes in Geophysics, 30, 183–193, https://doi.org/10.5194/npg-30-183-2023, 2023.

Power, T. and Berenson, D.: Variational Inference MPC using Normalizing Flows and Out-of-Distribution Projection, https://arxiv.org/abs/2205.04667, 2022.

Resseguier, V., Mémin, E., and Chapron, B.: Geophysical flows under location uncertainty, Part III SQG and frontal dynamics under strong turbulence conditions, Geophysical & Astrophysical Fluid Dynamics, 111, 209–227, https://doi.org/10.1080/03091929.2017.1312102, 2017.

Sawada, Y.: Quest for an efficient mathematical and computational method to explore optimal extreme weather modification, https://arxiv.org/abs/2405.08387, 2024a.

Sawada, Y.: Ensemble Kalman filter meets model predictive control in chaotic systems, SOLA, 20, 400–407, https://doi.org/10.2151/sola.2024-053, 2024b.

Schneider, T., Stuart, A. M., and Wu, J.-L.: Ensemble Kalman inversion for sparse learning of dynamical systems from time-averaged data, Journal of Computational Physics, 470, 111 559, https://doi.org/https://doi.org/10.1016/j.jcp.2022.111559, 2022.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and de Freitas, N.: Taking the Human Out of the Loop: A Review of Bayesian Optimization, Proceedings of the IEEE, 104, 148–175, 2016.

Sun, Q., Miyoshi, T., and Richard, S.: Control simulation experiments of extreme events with the Lorenz-96 model, Nonlinear Processes in Geophysics, 30, 117–128, https://doi.org/10.5194/npg-30-117-2023, 2023.

Vallis, G. K.: Atmospheric and Oceanic Fluid Dynamics: Fundamentals and Large-Scale Circulation, Cambridge University Press, 2 edn., 2017.

Williams, G., Drews, P., Goldfain, B., Rehg, J. M., and Theodorou, E. A.: Information-Theoretic Model Predictive Control: Theory and Applications to Autonomous Driving, IEEE Transactions on Robotics, 34, 1603–1622, https://doi.org/10.1109/TRO.2018.2865891, 2018.

Willoughby, H. E., Jorgensen, D. P., Black, R. A., and Rosenthal, S. L.: Project STORMFURY: A Scientific Chronicle 1962–1983, Bulletin of the American Meteorological Society, 66, 505 – 514, https://doi.org/10.1175/1520-0477(1985)066<0505:PSASC>2.0.CO;2, 1985.

Zhang, H., Mcfarquhar, G., Saleeby, S., and Cotton, W.: Impacts of Dust in the SAL as CCN on the Evolution of an Idealized Tropical Cyclone, AGU Fall Meeting Abstracts, 2006.

Appendix A: Standalone BO vs. vanilla MPPI

For completeness, we report here the comparison between standalone BO and vanilla MPPI in the same single-step Lorenz–96 setting as in Section 4.1.2. This appendix is included as a supplementary reference for the single-step setting. The parameter settings of vanilla MPPI are almost the same as those used for EKG-MPPI in Section 4.1.1. However, unlike EKG-MPPI, vanilla MPPI does not use EnKC-informed prior information for the intervention location or magnitude. Accordingly, the sampling distributions for the intervention location and intervention magnitude are fixed to $\mathcal{N}(0, 20.0)$ and $\mathcal{N}(0, 0.5)$, respectively. The parameter settings of BO are the same as those used in Section 4.1.2. As in Section 4.1.2, the experiments were repeated over 10 simulations with different random seeds, and we report the mean and standard deviation of each metric.

The results are summarized in Table A1. Vanilla MPPI yields fewer extreme events than standalone BO while requiring a comparable mean input magnitude. These results suggest that, even without EnKC-informed guidance, MPPI remains a competitive baseline in the present single-step setting.

Table A1. Comparison between vanilla MPPI and a standalone BO baseline in the single-step Lorenz–96 control setting. The second and third columns report the extreme-event count and the mean input magnitude, respectively, defined in the same manner as in Table 1. All values are shown as mean $\pm$ standard deviation over 10 simulations with different random seeds.

Method	Extreme-event count ($X \geq 12$)	Mean input magnitude
vanilla MPPI	6650 ± 138	0.335 ± 0.003
BO	8299 ± 255	0.340 ± 0.004

Appendix B: Rescaling used for visualization

To facilitate visual comparison in the two-dimensional performance–control plots, we rescale both the average intervention magnitude and the threshold-exceedance count (the horizontal-axis performance metric) by the corresponding mean over the five BO-MPPI runs for the same fixed value of λ. This rescaling is used only for visualization. Because each quantity is divided by a reference mean rather than mapped to the interval $[0, 1]$, we refer to the resulting quantities as *rescaled* (or *mean-scaled*) values rather than normalized values. All qualitative comparisons reported in the text are unchanged when the original, unscaled metrics are used. For each fixed λ, let $\mathcal{A} = \{100, 10, 1, 0.1, 0.01\}$. Let m_{EKG} and m_α denote the average intervention magnitudes obtained by EKG-MPPI and BO-MPPI, respectively. We define the BO-MPPI reference mean by

$$\bar{m}_{\mathrm{BO}} = \frac{1}{5} \sum_{\alpha \in \mathcal{A}} m_\alpha, \tag{B1}$$

and the corresponding rescaled quantities by

$$\tilde{m}_{\mathrm{EKG}} = \frac{m_{\mathrm{EKG}}}{\bar{m}_{\mathrm{BO}}}, \qquad \tilde{m}_\alpha = \frac{m_\alpha}{\bar{m}_{\mathrm{BO}}}. \tag{B2}$$

Similarly, let e_{EKG} and e_α denote the threshold-exceedance counts used on the horizontal axis. We define

$$\bar{e}_{\mathrm{BO}} = \frac{1}{5} \sum_{\alpha \in \mathcal{A}} e_\alpha, \tag{B3}$$

and

$$\tilde{e}_{\mathrm{EKG}} = \frac{e_{\mathrm{EKG}}}{\bar{e}_{\mathrm{BO}}}, \qquad \tilde{e}_\alpha = \frac{e_\alpha}{\bar{e}_{\mathrm{BO}}}. \tag{B4}$$

Values below 1 therefore indicates performance better than the BO-MPPI mean for the same λ on the corresponding axis. Smaller horizontal values indicate fewer threshold exceedances, and smaller vertical values indicate a smaller average intervention magnitude. Since each axis is rescaled by a common positive constant within each panel, this transformation does not change the within-panel ordering or Pareto dominance relations; it only changes the axis units for visualization.

Author contributions. **Haru Kuroki**: Conceptualization; investigation; methodology;validation; visualization; writing—original draft. **Kazumune Hashimoto**: Investigation; methodology; supervision; funding acquisition. **Yohei Sawada**: Investigation; methodology; supervision; funding acquisition. **Le Duc**: Investigation; methodology; **Masashi Minamide**: Investigation; writing—review and editing.

Competing interests. The authors declare no conflicts of interest.

Acknowledgements. This work was supported in part by the JST Moonshot R&D program under Grant JMPJMS2281.