ACUMEN: Active Cross-Entropy Method with Uncertainty-driven Neural ODEs for Data-Efficient System Identification in Healthcare

Amirhossein Afkhami Ardekani
Department of Mechanical Engineering
University of Alberta
Alberta, Canada
afkhamia@ualberta.ca

Mahmoud Zeydabadinezhad
Emory University
Atlanta, GA
mahmoud.zeydabadinezhad@emory.edu

Babak Mahmoudi
Emory University
Atlanta, GA
b.mahmoudi@emory.edu

Abstract

Building personalized, data-driven models of patient response to therapies with large parameter spaces, such as Deep Brain Stimulation, is a major challenge in precision medicine. We present ACUMEN, a data-efficient framework for digital twin–inspired modeling that couples Neural Ordinary Differential Equations (Neural ODEs) with uncertainty-driven planning via Cross-Entropy Method Model Predictive Control (CEM-MPC). An ensemble of Neural ODEs captures physiological dynamics, with ensemble disagreement quantifying epistemic uncertainty. CEM-MPC selects exploratory interventions under smoothness constraints, aided by optimistic state progression and adaptive scaling across state dimensions. In the Reinforcement Learning for Deep Brain Stimulation (RL-DBS) synthetic environment, ACUMEN reduces test error by up to 24.2% over passive data collection, while producing tighter uncertainty estimates. These results highlight ACUMEN's potential to lower sample complexity and enable safer, more personalized system identification in data-limited healthcare-inspired settings.

1 Introduction

Modeling of physiological dynamics in response to therapeutic interventions play a central role in designing personalized and adaptive therapies. Prior studies illustrate its broad impact, for example, glucose–insulin models have supported closed-loop control in type 1 diabetes [1], pharmacokinetic–pharmacodynamic modeling has been used to personalize anesthesia and sedation [2], and model-based control has guided neuromodulation strategies such as thalamic Deep Brain Stimulation (DBS) [3], a therapy requiring clinicians to navigate a vast parameter space to identify patient-specific settings. Neural ODEs have further been applied to capture tumor dynamics for survival prediction in oncology [4], while surrogate models have enabled reinforcement learning for closed-loop vagus nerve stimulation [5] and personalized interventions in mobile health settings [6]. Despite these successes, most approaches rely on large volumes of high-quality data and struggle to generalize to regions of the physiological state space where data are sparse. This limitation underscores the need for data-efficient, uncertainty-aware system identification methods that can adaptively guide exploration in healthcare applications.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Learning from Time Series for Health.

A promising strategy to address this challenge is active learning, where the model does not passively rely on uniformly collected data but instead actively guides data acquisition toward the most informative regions of the state space. In this setting, a surrogate model such as a neural network can be trained to approximate the underlying dynamics, while its epistemic uncertainty is quantified to highlight areas where predictions are unreliable. By deliberately sampling in these high-uncertainty regions, the model is refined in a targeted and data-efficient manner, avoiding redundant exploration of well-understood states. Prior work has demonstrated the effectiveness of such approaches in improving robustness and sample efficiency for learning dynamical systems [7, 8]. Moreover, advances in uncertainty estimation [9, 10, 11] and adaptive experimental design [12] provide a principled basis for steering exploration. In healthcare applications, these uncertainty-driven strategies are particularly valuable since they enable patient-specific models to be refined with minimal interventions, thereby addressing both ethical constraints and practical limitations in clinical data collection [13].

In this work, we introduce ACUMEN (Active Cross-Entropy Method with Uncertainty-driven Neural ODEs), a framework designed to advance data-driven digital twin modeling through uncertainty-aware active learning on time-series data, tested for DBS. Collecting high-quality physiological signals, ranging from neural recordings to wearable biosensors, is often challenging due to noise, heterogeneity, and practical or ethical limits on data acquisition. This makes efficient and principled exploration strategies essential for building reliable patient-specific models. ACUMEN addresses this challenge by combining a Neural ODE–based surrogate model with principled epistemic uncertainty quantification and an uncertainty-seeking exploration strategy based on the Cross-Entropy Method for Model Predictive Control (CEM-MPC). By actively prioritizing underrepresented or uncertain trajectories, ACUMEN accelerates model refinement while minimizing redundant sampling. To demonstrate the approach, we evaluate ACUMEN in a synthetic neuromodulation-inspired environment, the RL-DBS testbed, which simulates ensembles of interacting neurons whose pathological synchrony underlies neurological disorders such as Parkinson's disease. This environment provides nonlinear and noise-sensitive dynamics resembling neural recordings, allowing us to test uncertainty-driven active learning in a controlled setting. More broadly, ACUMEN illustrates how bridging uncertainty-driven active learning with continuous-time modeling can yield scalable methods for healthcare time series, moving toward a new generation of personalized, data-efficient therapies.

2 ACUMEN Algorithm

2.1 Problem Formulation

We formulate system identification in healthcare as learning patient-specific dynamical models from irregular and noisy time-series. Let $x(t) \in \mathbb{R}^n$ denote the latent state (e.g., neural or cardiovascular activity), and $u(t) \in \mathbb{R}^d$ denote external interventions (e.g., stimulation, drug dosage). Observations are collected at irregular times $\{t_i\}_{i=1}^N$:

$$y_i = h_\phi(x(t_i)) + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, \sigma^2), \tag{1}$$

with latent dynamics governed by $\dot{x}(t) = f_\theta(x(t), u(t), t)$, the goal is to predict future trajectories under interventions while minimizing unnecessary, unsafe, or expensive exploration.

2.2 ACUMEN Overview

We propose ACUMEN, an iterative framework that combines continuous-time modeling, principled uncertainty estimation, and active exploration. In each iteration, CEM-MPC generates interventions that maximize epistemic uncertainty, the Neural ODE ensemble is retrained on the expanded dataset, and adaptive scaling normalizes uncertainty estimates across heterogeneous variables. Repeating this cycle progressively refines patient-specific models while minimizing redundant sampling (Fig. 1).

2.3 Neural ODE–based System Identification

We use Neural ODEs [14] to model continuous-time dynamics. Given the current state x_t and control input u_t, the dynamics evolve as $\dot{x}(t) = f_\theta(x(t), u(t), t)$, with solutions obtained via numerical integration using a Runge–Kutta (RK4) solver, $x_{t+\Delta t} = \mathrm{ODESolve}(x_t, f_\theta, u_t, [0, \Delta t])$. Here f_θ is parameterized by a MultiLayer Perceptron (MLP), and training minimizes Mean Squared Error

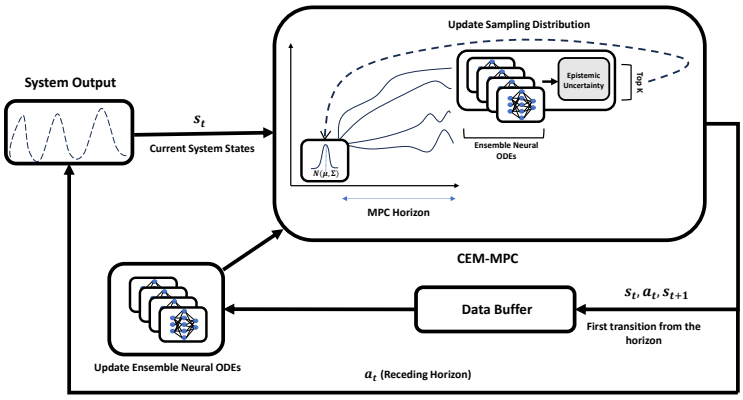

Figure 1: Overview of the ACUMEN algorithm. From the current state s_t, CEM-MPC samples and evaluates candidate action sequences with an ensemble of Neural ODEs, guided by epistemic uncertainty. The distribution is updated using elite sequences, the first action a_t is executed, and the transition (s_t, a_t, s_{t+1}) is stored in the buffer. The ensemble is then retrained, completing a cycle of uncertainty-driven exploration, data collection, and model refinement.

(MSE) between predicted and observed next states. This continuous-time formulation provides stable rollouts and can flexibly capture nonlinear state transitions.

2.4 Uncertainty Estimation

To guide exploration, ACUMEN estimates epistemic uncertainty via an ensemble of Neural ODEs. For M ensemble members, the predictive mean and variance are computed as

$$\hat{x}(t) = \frac{1}{M} \sum_{m=1}^{M} \mu_m(x, u, t), \quad \sigma^2_{\text{epi}}(x) = \frac{1}{M} \sum_{m=1}^{M} \mu_m(x, u, t)^2 - \hat{x}(t)^2. \tag{2}$$

This variance reflects model disagreement and highlights underrepresented regions of the state space. The epistemic uncertainty is considered in our method, since it is sufficient for directing exploration toward poorly understood dynamics.

2.5 Uncertainty-Driven Exploration with CEM-MPC

We adapt CEM-MPC [15, 16] to maximize epistemic uncertainty. For a candidate action sequence $a_{t:t+H-1}$, the exploration score is

$$J_{\text{unc}}(a_{t:t+H-1}) = \sum_{h=0}^{H-1} w_h \left[\sum_{d} \sigma^2_{\text{epi},d}(x_h) \right] - \lambda_{\text{du}} \sum_{h=1}^{H-1} \|a_{t+h} - a_{t+h-1}\|^2, \tag{3}$$

where $\sigma^2_{\text{epi},d}(x_h)$ is the ensemble variance for state dimension d at step h, and λ_{du} penalizes abrupt changes between consecutive actions to encourage smooth control sequences. Only the first action of the best sequence is executed before replanning. Full pseudocode and implementation details are provided in Appendix A.

3 Experimental Setup

We evaluate ACUMEN in the Reinforcement Learning for Deep Brain Stimulation (RL-DBS) environment [17], a physiologically motivated testbed that emulates ensembles of interacting neurons whose pathological synchrony underlies neurological disorders such as Parkinson's disease. External stimulation acts as uniform suppression pulses, and the observable state is the mean-field activity of the population, where successful control corresponds to desynchronization. This setup provides

synthetic, nonlinear, and noise-sensitive time-series resembling neural recordings, making it suitable for testing uncertainty-driven active learning algorithms.

ACUMEN is trained on data collected through an iterative cycle of seeding, uncertainty-guided exploration, and retraining. An ensemble of Neural ODEs models the system dynamics, with epistemic uncertainty from model disagreement guiding exploration through CEM-MPC. Across 40 episodes of 100 steps each, the algorithm collects approximately 4,500 transitions in total, progressively refining the learned dynamics. Additional details on the data collection protocol, architecture, and hyperparameters are provided in Appendix B.

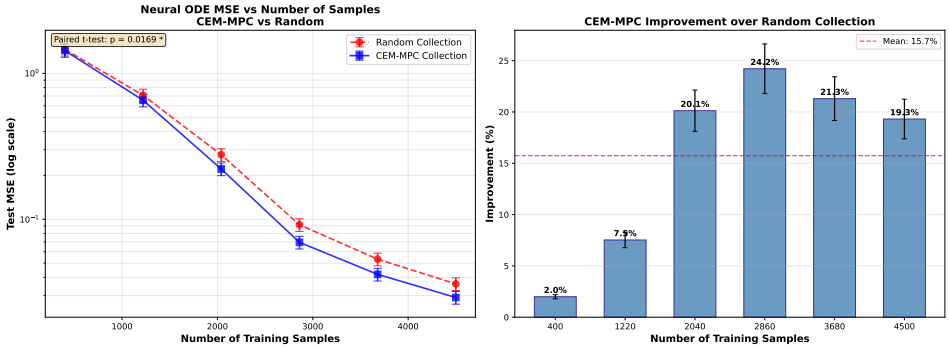

Figure 2: Comparison of data collection strategies with statistical validation. Left: Test MSE of Neural ODE ensembles as a function of the number of training samples, plotted on a logarithmic scale. Models trained on CEM-MPC collected data consistently achieve lower error than those trained on randomly collected data (paired t-test: $p = 0.017$, statistically significant). Right: Percentage improvement of CEM-MPC over random sampling at each dataset size, showing consistent gains across all sample sizes with an average improvement of $15.7\% \pm 8.8\%$. The dashed red line indicates the mean improvement across all conditions.

4 Results and Discussion

We compared Neural ODE ensembles trained on datasets collected with uncertainty-driven CEM-MPC exploration against ensembles trained on equally sized datasets from random actions in the RL-DBS environment. To ensure fairness, models were trained on progressively larger subsets of each dataset, with optimization steps scaled to dataset size.

Figure 2 demonstrates that both strategies reduce error as more data are acquired, but CEM-MPC consistently achieves statistically significant lower test MSE (paired t-test: $p = 0.017$). The improvement ranges from 2.0% to 24.2% across different sample sizes, with an overall mean improvement of $15.7\% \pm 8.8\%$. Notably, CEM-MPC outperformed random sampling in every single comparison (Wilcoxon signed-rank test: $p = 0.031$), indicating robust and consistent superiority. The improvement is modest with fewer than 1000 samples but becomes more substantial with larger datasets, demonstrating that uncertainty-guided exploration generates more informative data per sample and significantly reduces the sample complexity of system identification.

To validate the statistical significance, we conducted comprehensive bootstrap analysis and consistency testing. Figure 4 presents detailed statistical validation showing that the observed improvements are robust and statistically significant. Bootstrap resampling (10,000 iterations) confirms that the mean difference in MSE is reliably positive with a 95% confidence interval of [0.015, 0.045], which excludes zero. Furthermore, CEM-MPC outperformed random sampling consistently across all six sample sizes tested, with no exceptions, demonstrating the robustness of the uncertainty-guided approach across different data regimes. Additional results are provided in Appendix C.

5 Conclusion

We presented ACUMEN, a framework that combines Neural ODE ensembles with uncertainty-guided exploration via CEM-MPC for data-efficient system identification in healthcare. By targeting

underrepresented regions of the state space, ACUMEN reduces sample complexity and improves calibration, achieving up to 24% lower test error in the RL-DBS environment compared to random data collection. These results highlight the value of coupling continuous-time modeling with uncertainty-aware active learning. Future work will extend ACUMEN to real physiological data and closed-loop control applications.

Acknowledgments and Disclosure of Funding

This work was supported by Google Summer of Code. The authors declare no competing interests.

References

[1] S. Schmidt, D. Boiroux, A. K. Duun-Henriksen, L. Frøssing, O. Skyggebjerg, J. B. Jørgensen, N. K. Poulsen, H. Madsen, S. Madsbad, and K. Nørgaard. Model-based closed-loop glucose control in type 1 diabetes: the diacon experience. *Journal of Diabetes Science and Technology*, 7(5):1255–1264, 2013.

[2] D. J. Eleveld, P. Colin, A. R. Absalom, and M. M. Struys. Pharmacokinetic–pharmacodynamic model for propofol for broad application in anaesthesia and sedation. *British Journal of Anaesthesia*, 120(5):942–959, 2018.

[3] Y. Tian, S. Saradhi, E. Bello, M. D. Johnson, G. D'Eleuterio, M. R. Popovic, and M. Lankarany. Model-based closed-loop control of thalamic deep brain stimulation. *Frontiers in Network Physiology*, 4:1356653, 2024.

[4] M. Laurie and J. Lu. Explainable deep learning for tumor dynamic modeling and overall survival prediction using neural-ode. *NPJ Systems Biology and Applications*, 9(1):58, 2023.

[5] P. Sarikhani, H.-L. Hsu, M. Zeydabadinezhad, Y. Yao, M. Kothare, and B. Mahmoudi. Reinforcement learning for closed-loop regulation of cardiovascular system with vagus nerve stimulation: a computational study. *Journal of Neural Engineering*, 21(3):036027, 2024.

[6] M. El Mistiri, O. Khan, C. A. Martin, E. Hekler, and D. E. Rivera. Data-driven mobile health: System identification and hybrid model predictive control to deliver personalized physical activity interventions. *IEEE Open Journal of Control Systems*, 4:83–102, 2025.

[7] V. L. Nguyen, S. Destercke, and E. Hüllermeier. Epistemic uncertainty sampling. In *International Conference on Discovery Science*, pages 72–86, Cham, October 2019. Springer International Publishing.

[8] A. Saviolo, J. Frey, A. Rathod, M. Diehl, and G. Loianno. Active learning of discrete-time dynamics for uncertainty-aware model predictive control. *IEEE Transactions on Robotics*, 40:1273–1291, 2023.

[9] Y. Gal, R. Islam, and Z. Ghahramani. Deep bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1183–1192. PMLR, 2017.

[10] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059. PMLR, 2016.

[11] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates Inc., 2017.

[12] K. Chaloner and I. Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304, 1995.

[13] O. Gottesman, F. Johansson, M. Komorowski, A. Faisal, D. Sontag, F. Doshi-Velez, and L. A. Celi. Guidelines for reinforcement learning in healthcare. *Nature Medicine*, 25(1):16–18, 2019.

[14] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, pages 6571–6583, 2018.

[15] H. Bharadhwaj, K. C. Xie, and F. Shkurti. Model-predictive control via cross-entropy and gradient-based optimization. In *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, pages 605–615. PMLR, 2020.

[16] C. Pinneri, S. Sawant, S. Blaes, and G. Martius. Sample-efficient cross-entropy method for real-time planning. In *Proceedings of the 4th Conference on Learning for Dynamics and Control*, pages 389–402. PMLR, 2021.

[17] Dmitrii Krylov, Remi Tachet, Romain Laroche, Michael Rosenblum, and Dmitry V. Dylov. Reinforcement learning framework for deep brain stimulation study. *arXiv preprint arXiv:2002.10948*, 2020.

A Methodological Details and Pseudocode

A.1 Neural ODE Training Details

Each ensemble member is implemented as a Neural ODE with dynamics parameterized by an MLP. The MLP receives the current state x_t and action u_t and outputs the instantaneous derivative $\dot{x}(t)$. State updates are computed with a RK4 solver. Training minimizes the MSE between predicted and observed next states, using Adam optimization with gradient clipping for stability.

A.2 Uncertainty Estimation

To quantify epistemic uncertainty in ACUMEN, we employ an ensemble of independently trained Neural ODE models. Each ensemble member is parameterized by an MLP that receives the current state x and action u and outputs the instantaneous derivative $\dot{x}(t)$. The models are trained on the same replay buffer but initialized with different random seeds and mini-batch orderings, which induces diversity in their learned representations of the underlying dynamics.

Given M ensemble members, the predictive mean trajectory at time t is obtained as the average output across all members:

$$\hat{x}(t) = \tfrac{1}{M} \sum_{m=1}^{M} \mu_m(x, u, t).$$

The epistemic variance is then defined as the variance of the predictions around this mean:

$$\sigma^2_{\text{epi}}(x) = \tfrac{1}{M} \sum_{m=1}^{M} \mu_m(x, u, t)^2 - \hat{x}(t)^2.$$

This variance directly measures model disagreement, high variance indicates regions of the state space where ensemble members diverge due to insufficient or ambiguous training data, while low variance corresponds to well-explored and consistently modeled dynamics.

Notably, only epistemic uncertainty is modeled in this work. Aleatoric uncertainty (noise inherent to the measurements) is not explicitly parameterized, as our primary objective is to identify poorly understood regions of the dynamical system where additional data would be most informative. To prevent bias toward state dimensions with naturally larger magnitudes, the raw uncertainty values are normalized using adaptive scaling factors computed from replay buffer statistics.

These uncertainty estimates serve as the key driver for exploration in ACUMEN: the CEM-MPC actively prioritizes interventions that maximize σ^2_{epi}, thereby steering the data collection process toward underrepresented trajectories. This mechanism ensures that model refinement is targeted and data-efficient, avoiding redundant sampling in well-characterized regimes of the system.

A.3 CEM-MPC Algorithm

CEM-MPC is adapted to prioritize exploration in high-uncertainty regions. At each iteration, action sequences are sampled from a Gaussian distribution, evaluated by rolling out the ensemble models, and scored based on their cumulative epistemic uncertainty. We include a smoothness penalty on consecutive actions to avoid abrupt changes and an optimism parameter β that biases rollouts toward uncertain directions. After scoring, the distribution parameters (mean and variance) are updated using elite samples weighted by softmax scores. Only the first action of the highest-scoring sequence is executed before replanning.

A.4 Enhanced Exploration Strategies

Optimistic progression biases rollouts toward uncertain regions:

$$x^{\text{opt}}_{h+1} = \mu_{h+1} + \beta \sqrt{\sigma_{\text{epi}}(x_h)} \cdot \epsilon_h, \tag{4}$$

with $\epsilon_h \sim \mathcal{N}(0, I)$. This encourages the planner to sample trajectories that probe uncertain directions of the state space. In our method, we used an optimism-based exploration.

A.5 Adaptive Scaling and Training the ensemble Neural ODE

To avoid bias toward high-variance dimensions, epistemic uncertainty is normalized:

$$\tilde{\sigma}_{\text{epi}}(x) = \frac{\sigma_{\text{epi}}(x)}{\sigma_{\text{scale}} + \epsilon}, \tag{5}$$

where σ_{scale} is the empirical standard deviation estimated from the replay buffer. Each ensemble member is retrained periodically using MSE loss:

$$\mathcal{L}_m = \frac{1}{N} \sum_{i=1}^{N} \| f_{\theta_m}(x_i, u_i, t_i) - x_{i+1} \|_2^2. \tag{6}$$

A.6 Complete Algorithm

For completeness, the full ACUMEN loop is summarized in Algorithm 1, while the uncertainty-driven CEM-MPC exploration strategy is detailed in Algorithm 2. These algorithms work in conjunction to provide adaptive system identification through targeted data collection in regions of high epistemic uncertainty.

Algorithm 1 ACUMEN: Active Cross-Entropy Method with Uncertainty-driven Neural ODEs

Require: Environment, ensemble size M, episodes E, steps per episode S
Initialize ensemble $\{f_{\theta_m}\}_{m=1}^{M}$ of Neural ODEs, Adam optimizers, replay buffer $\mathcal{D}$
Initialize running state scale tracker σ_{scale}
Collect N_{seed} transitions with random actions $\rightarrow \mathcal{D}$
Train ensemble on $\mathcal{D}$ for T_{init} steps with MSE loss
Set the last collected data from the environment as x_{current}
for episode $e = 1, \ldots, E$ **do**
  Update state scaling σ_{scale} from current $\mathcal{D}$ (subsampled if $|\mathcal{D}| > 5000$)
  Set σ_{scale} in CEM-MPC policy
  for step $s = 1, \ldots, S$ **do**
    if random() $< p_{\text{noise}}$ **then**
      $a \leftarrow$ random action from action space for exploration
    else
      $a \leftarrow$ CEM-MPC(x_{current}, ensemble, uncertainty-seeking objective)
    end if
    Clip action: $a \leftarrow \text{clip}(a, a_{\min}, a_{\max})$
    Execute action a, observe transition $(x_{\text{current}}, a, x')$
    Add $(x_{\text{current}}, a, x')$ to replay buffer $\mathcal{D}$
    Update current state $x_{\text{current}} \leftarrow x'$
  end for
  Compute adaptive retrain steps: $T_{\text{retrain}} \leftarrow \min(300, T_{\text{base}} + \lfloor |\mathcal{D}|/500 \rfloor)$
  Retrain ensemble on $\mathcal{D}$ for T_{retrain} steps with gradient clipping
end for
Save dataset $\mathcal{D}$
return Trained ensemble $\{f_{\theta_m}\}_{m=1}^{M}$

Algorithm 2 CEM-MPC for Uncertainty-Driven Exploration

Require: Current state x_{current}, ensemble $\{f_{\theta_m}\}_{m=1}^{M}$, horizon H, iterations I, sequences N, elites K

Initialize mean $\mu \leftarrow \mathbf{0}_{H \times d_a}$, std $\sigma \leftarrow \sigma_{\text{init}} \cdot \text{action\_range}$
Set time integration vector $t \leftarrow [0, \Delta t]$ for ODE solver
Initialize minimum std: $\sigma_{\min} \leftarrow \sigma_{\text{min\_frac}} \cdot \text{action\_range}$
for iteration $i = 1, \ldots, I$ **do**
  Sample N action sequences: $\{a_{0:H-1}^{(j)}\}_{j=1}^{N} \sim \mathcal{N}(\mu, \text{diag}(\sigma^2))$
  Clip actions to bounds: $a_{0:H-1}^{(j)} \leftarrow \text{clip}(a_{0:H-1}^{(j)}, a_{\min}, a_{\max})$
  for sequence $j = 1, \ldots, N$ **do**
    Initialize score $S_j \leftarrow 0$, state $x_0^{(j)} \leftarrow x_{\text{current}}$, discount $\gamma \leftarrow 1$, prev_action $\leftarrow$ None
    for horizon step $h = 0, \ldots, H-1$ **do**
      Get action $a_h^{(j)}$ from sequence $a_{0:H-1}^{(j)}$
      Compute epistemic uncertainty: $\mu_{h+1}, \sigma^2_{\text{epi}}(x_h^{(j)}) \leftarrow$ EXTRACTUNCERTAINTIES(ensemble, $x_h^{(j)}, a_h^{(j)}, t$)
      Compute information gain:
      info $\leftarrow \sum_d \log(\sigma^2_{\text{epi},d}(x_h^{(j)}) + \epsilon)$
      Compute action smoothness penalty: $du \leftarrow 0$ if $h = 0$ else $\|a_h^{(j)} - a_{h-1}^{(j)}\|_2^2$
      Compute step score: $s_h \leftarrow \text{info} - \lambda_{\text{du}} \cdot du$
      Update total score: $S_j \leftarrow S_j + \gamma \cdot s_h$
      Update discount: $\gamma \leftarrow \gamma \cdot \gamma_{\text{discount}}$
      Store previous action: prev_action $\leftarrow a_h^{(j)}$
      Optimistic rollout: $x_{h+1}^{(j)} \leftarrow \mu_{h+1} + \beta\sqrt{\sigma^2_{\text{epi}}(x_h^{(j)})} \cdot \text{sign}(\mathcal{N}(0, I))$
    end for
  end for
  Select top K elite sequences: elites $\leftarrow \arg\text{topk}(\{S_j\}, K)$
  Get elite scores: vals $\leftarrow \{S_j : j \in \text{elites}\}$
  Compute elite weights: $w_j \leftarrow \text{softmax}(\text{vals} - \max(\text{vals}))$
  Update mean: $\mu \leftarrow \frac{\sum_{j \in \text{elites}} w_j \cdot a_{0:H-1}^{(j)}}{\sum_{j \in \text{elites}} w_j + \epsilon}$
  Update variance: $\sigma^2 \leftarrow \frac{\sum_{j \in \text{elites}} w_j \cdot (a_{0:H-1}^{(j)} - \mu)^2}{\sum_{j \in \text{elites}} w_j + \epsilon}$
  Update std: $\sigma \leftarrow \sqrt{\sigma^2 + \epsilon}$
  Clamp std: $\sigma \leftarrow \text{clamp}(\sigma, \sigma_{\min}, \text{action\_range})$
end for
Shift mean and std for next timestep: $\mu \leftarrow [\mu_{1:H-1}, \mathbf{0}]$, $\sigma \leftarrow \max([\sigma_{1:H-1}, \sigma_{H-1}], \sigma_{\min})$
return First action: $a^* \leftarrow \mu[0]$

B Experimental Details

B.1 Data Collection Protocol

The experiment begins with a seeding phase where 500 random transitions are collected to initialize the ensemble training. Following this, an initial training phase updates all ensemble members for 300 steps using MSE loss.

The active learning phase then consists of 40 episodes, each collecting 100 new transitions through CEM-MPC guided exploration. At each step, the algorithm either executes a random action with a fixed probability to maintain exploration diversity or selects an action using the uncertainty-seeking CEM-MPC policy. The policy evaluates candidate action sequences via ensemble rollouts and scores them based on epistemic uncertainty, with optimistic state progression encouraging exploration toward uncertain regions. After each episode, the ensemble is retrained on the expanded replay buffer with steps proportional to buffer size, and replay buffer statistics are used for adaptive scaling. The full process yields approximately 4,500 transitions in total.

B.2 Model Architecture and Training

The Neural ODE ensemble consists of 5 independent models, each parameterized by a 3-layer MLP with 128 hidden units and Tanh activations. The ODE solver uses a RK4 integration with step size 0.5. Models are trained with Adam (learning rate 10^{-3}, batch size 256) and gradient clipping with norm 1.0 to ensure stability. The replay buffer has capacity 100k transitions.

B.3 Environment Configuration

The RL-DBS oscillator environment provides a continuous state space of shape $(2, L)$, where the two dimensions correspond to mean-field activity in the x and y-directions, and L is the history length (e.g., $L = 8$ in our experiments). This structure represents neural population dynamics through parallel channels of activity. The action space is one-dimensional and continuous, corresponding to external stimulation control. Pathological regimes correspond to sustained synchronous oscillations of the mean-field activity, while successful control is defined as desynchronization toward a stable equilibrium.

The environment dynamics exhibit complex nonlinear behavior, as illustrated in Figure 3. During baseline periods (no intervention), the system displays characteristic oscillatory patterns in both state dimensions, reflecting the underlying pathological synchronization. The RL-DBS intervention modulates these dynamics through continuous control actions, demonstrating the system's responsiveness to external stimulation. The phase space representation reveals the natural attractor structure of the environment, showing how the system evolves through different dynamical regimes under varying control conditions.

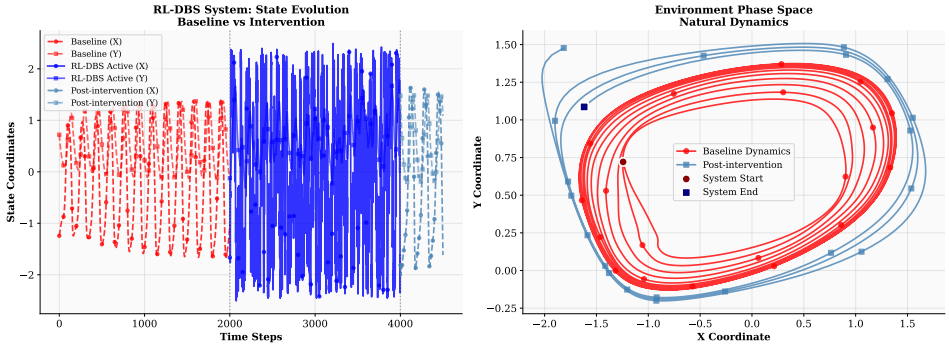

Figure 3: RL-DBS environment dynamics demonstration. Left: temporal evolution of state coordinates (X and Y) showing baseline behavior, RL-DBS intervention period, and post-intervention dynamics. The system exhibits characteristic oscillatory patterns during baseline periods, with modified dynamics during active stimulation. Right: phase space representation of the natural system dynamics, illustrating the environment's attractor structure and typical trajectory patterns in the absence of intervention. The natural dynamics reveal the underlying pathological synchronization that RL-DBS aims to control.

Importantly, the RL-DBS testbed serves as a proxy for clinical reality. Acquiring high-quality physiological data from real patients undergoing deep brain stimulation is challenging due to ethical constraints, patient safety concerns, and the limited availability of invasive neural recordings. By contrast, this synthetic environment provides a safe and reproducible platform for experimentation, while still capturing key difficulties of real-world physiological modeling: strong nonlinear dynamics, noisy measurements, and heterogeneous responses to interventions. Framing the RL-DBS testbed in this way positions synthetic-only validation not as a limitation, but as a principled and necessary step toward future clinical translation.

B.4 CEM-MPC Hyperparameter and Training Details

The CEM-MPC hyperparameters in Table 1 require additional explanation as they directly control the uncertainty-driven exploration behavior. The horizon of 35 steps determines how far into the future the algorithm plans when evaluating action sequences, balancing computational cost with

the ability to discover long-term uncertain trajectories. At each planning step, CEM samples 800 sequences of candidate actions and evaluates them through ensemble rollouts, then selects the top 80 elite sequences (10% of candidates) with the highest uncertainty-based scores. This elite fraction balances exploitation of promising regions with sufficient diversity to avoid premature convergence. The algorithm iteratively refines the action distribution through 8 iterations, allowing sufficient convergence while maintaining computational tractability. The initial standard deviation fraction of 0.40 determines the breadth of initial exploration relative to the action space range, starting with relatively wide sampling to ensure adequate coverage, while the minimum standard deviation fraction of 0.10 prevents the distribution from collapsing too aggressively, maintaining exploratory diversity throughout the search. In addition, a smoothness penalty on consecutive actions (λ_{du}) discourages abrupt control changes, and an optimism parameter (β) biases rollouts toward uncertain directions. Together, these parameters enable CEM-MPC to effectively balance between focused exploitation of high-uncertainty regions and broad exploration of the action space.

Table 1: Experimental hyperparameters for ACUMEN evaluation

Parameter	**Value**
Model Architecture	
Ensemble size	5
Hidden dimensions	128
MLP layers	3
Activation function	Tanh
ODE solver	RK4
ODE step size	0.5
Training	
Learning rate	1e-3
Optimizer	Adam
Batch size	256
Gradient clipping	1.0
Loss function	MSE
Data Collection	
Seed steps	500
Episodes	40
Steps per episode	100
Total transitions	$\sim$4,500
Initial training steps	300
Retraining steps	400
Action noise probability	0.2
Replay buffer capacity	100,000
CEM-MPC	
Horizon	35
Iterations	8
Sequences	800
Elites	80
Initial std fraction	0.40
Minimum std fraction	0.10
Discount factor (γ)	0.98
Action smoothness penalty (λ_{du})	0.05
Optimism parameter (β)	1.5

C Additional Experimental Results

C.1 Statistical Validation of Results

To validate the superiority of CEM-MPC over random sampling, we conducted bootstrap analysis and consistency testing.

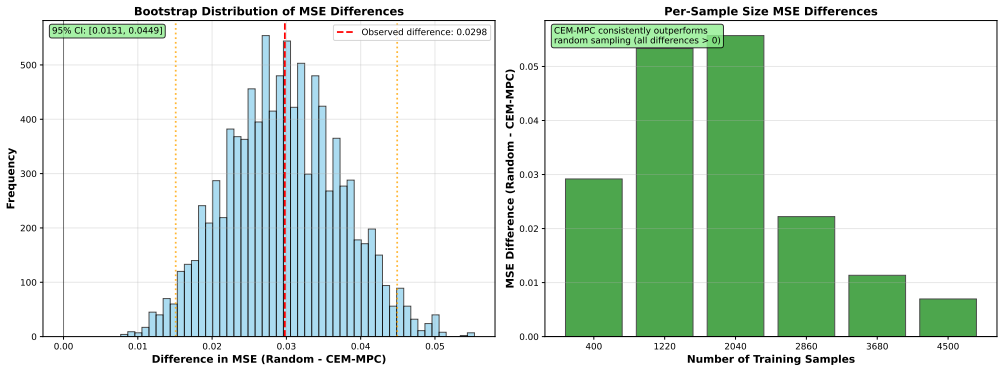

Figure 4: Statistical validation of CEM-MPC superiority. Left: Bootstrap distribution of MSE differences (Random - CEM-MPC) from 10,000 resampling iterations, showing the observed mean difference (red dashed line) and 95% confidence interval (orange dotted lines). The distribution is entirely positive, confirming statistical significance. Right: Per-sample size MSE differences demonstrating that CEM-MPC consistently outperforms random sampling across all tested dataset sizes, with no exceptions.

The bootstrap distribution demonstrates remarkable consistency in the performance advantage, with the narrow spread around the mean difference indicating low variance in the improvement across different data subsamples, suggesting the benefits are not driven by outliers or specific data characteristics, while the symmetric shape of the distribution further validates the robustness of the statistical test. The per-sample consistency analysis provides granular evidence of systematic superiority, showing that even at the smallest dataset size (500 samples) CEM-MPC maintains a positive advantage, and that this benefit grows substantially as more data becomes available, suggesting that uncertainty-guided exploration becomes increasingly effective as the model accumulates sufficient experience to identify truly informative regions. Collectively, these comprehensive statistical tests eliminate concerns about random chance, measurement noise, or dataset-specific artifacts, firmly establishing that the uncertainty-driven approach provides reliable and reproducible improvements in system identification tasks.

C.2 Uncertainty Calibration Analysis

To further evaluate model reliability, we analyzed how data collection strategies affect predictive uncertainty and accuracy. Figure 5 presents two complementary views. The left panel shows per-dimension average uncertainty across state channels, directly summarizing the detailed trajectory-level results in Figure 6. Ensembles trained on CEM-MPC data consistently exhibit lower uncertainty than those trained on random data, indicating that active exploration improves calibration across individual state dimensions. The right panel compares test error (MSE) between the two methods, showing that CEM-MPC achieves substantially lower predictive error (0.002046 vs 0.002640), corresponding to a 22.5% relative improvement.

Together, these results demonstrate that uncertainty-aware exploration improves both calibration and predictive accuracy: the left panel captures reductions in channel-level uncertainty, while the right panel highlights system-level gains in accuracy.

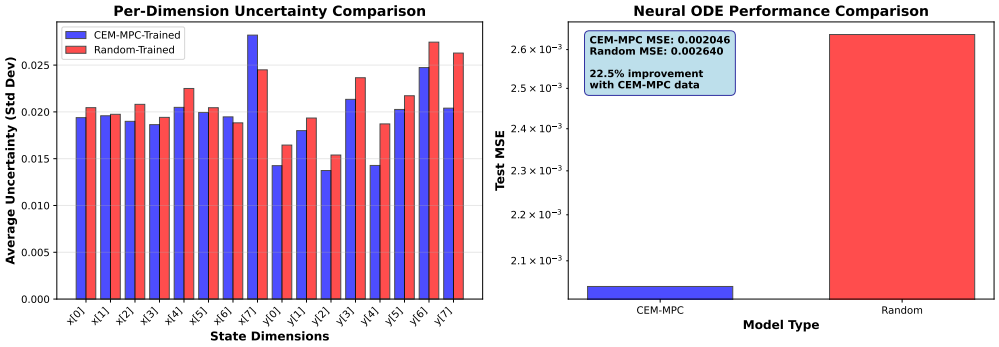

Figure 5: Uncertainty and error summary. Left: Per-dimension average uncertainty (ensemble standard deviation) for CEM-MPC–trained vs random-trained models, showing consistent reductions with CEM-MPC across most channels. Right: Comparison of test error (MSE), where the CEM-MPC model attains lower error (0.002046 vs 0.002640) and a 22.5% relative improvement.

C.3 Additional Trajectory-Level Results

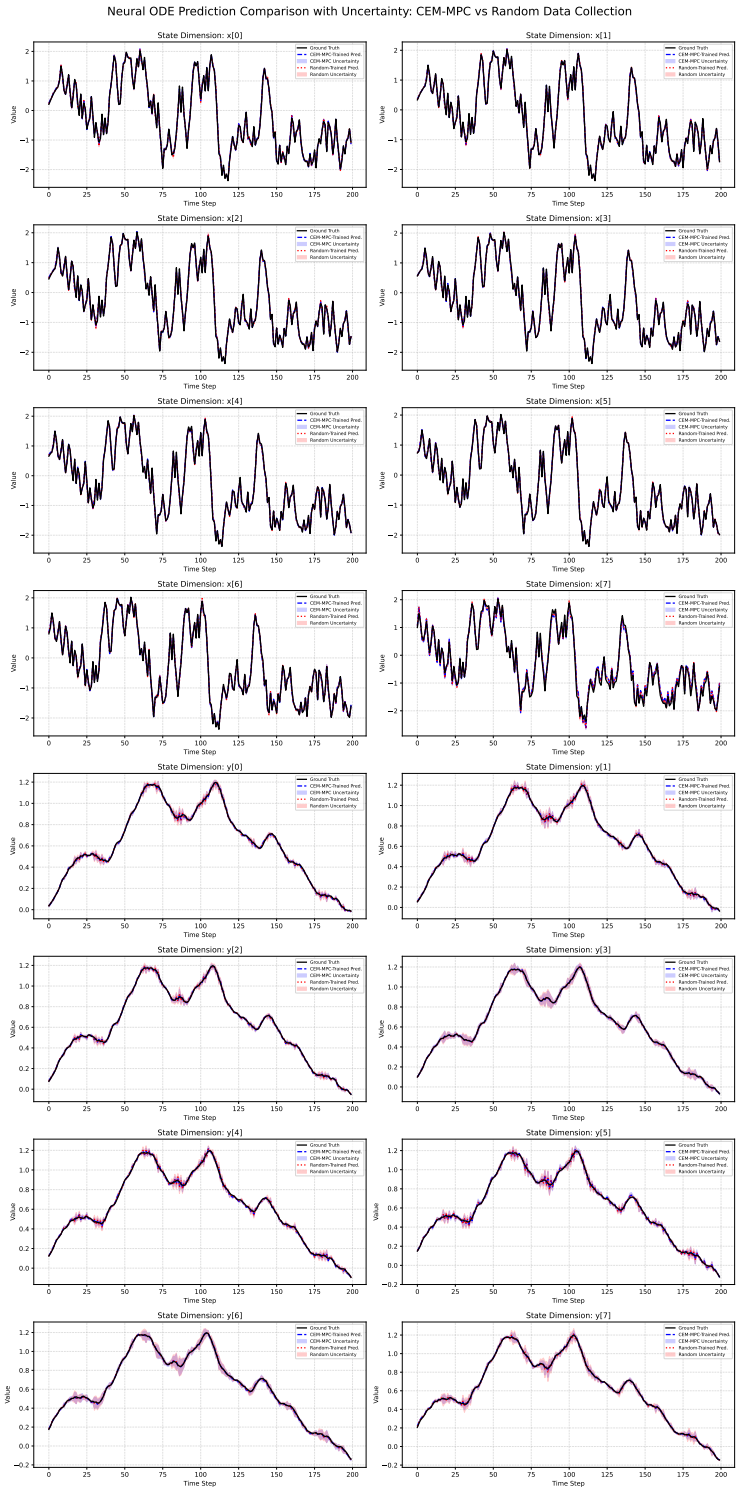

Figure 6: Trajectory predictions with per-channel uncertainty. each subplot shows one state dimension over the first 200 time steps: ground truth, ensemble mean predictions from models trained on CEM-MPC vs random data, and their shaded uncertainty bands (ensemble standard deviation). CEM-MPC training produces closer tracking and narrower uncertainty envelopes across many channels.