Consensus-based Optimization (CBO) for trajectory optimization in Robotics

Xudong Sun*, Armand Jordana†, Massimo Fornasier*‡, Jalal Etesami*, Majid Khadiv*
*Technical University of Munich (TUM), Germany
†LAAS-CNRS, France
‡Munich Center for Machine Learning (MCML), Munich, Germany

***Abstract*—Zero-order optimization has recently received significant attention for designing optimal trajectories and policies for robotic systems. However, most existing methods (e.g., MPPI, CEM, and CMA-ES) are local in nature, as they rely on gradient estimation. In this paper, we introduce consensus-based optimization (CBO) to robotics, which is guaranteed to converge to a global optimum under mild assumptions. We provide theoretical analysis and illustrative examples that give intuition into the fundamental differences between CBO and existing methods. To demonstrate the scalability of CBO for robotics problems, we consider three challenging trajectory optimization scenarios: (1) a long-horizon problem for a simple system, (2) a dynamic balance problem for a highly underactuated system, and (3) a high-dimensional problem with only a terminal cost. Our results show that CBO is able to achieve lower costs with respect to existing methods on all three challenging settings. This opens a new framework to study global trajectory optimization in robotics. (This is a short version of [1] to appear at RSS 2026).**

***Index Terms*—consensus-based optimization, trajectory optimization, global optimization, robotics, zero-order methods**

I. Introduction

Trajectory optimization in robotics remains difficult when the cost landscape is highly nonconvex, the horizon is long, and contact interactions make gradients brittle or expensive to obtain. Recent parallel simulation tools have made zero-order optimization increasingly practical because many candidate control trajectories can be evaluated simultaneously without requiring a differentiable simulator [2, 3, 4, 5]. This has made methods such as model predictive path integral control (MPPI), the cross-entropy method (CEM), and CMA-ES standard tools for contact-rich planning and control.

At the same time, the dominant zero-order methods used in robotics still rely on refining a local sampling distribution around the current estimate [6, 7, 8]. This becomes fragile in settings with narrow feasible corridors, long horizons, or multiple distant basins of attraction (local optimizers). In those cases, the optimization may stall prematurely even with many rollouts.

This workshop paper investigates consensus-based optimization (CBO) as a practical alternative for global trajectory optimization in robotics [9, 10, 11]. Instead of repeatedly resampling from a single local distribution, CBO evolves a population of particles whose motion is coupled through a cost-weighted consensus point. This creates a simple mechanism for combining global exploration with progressive concentration around promising solutions.

The contributions of this workshop paper are:

- We present a compact robotics-oriented view of CBO that emphasizes its search dynamics in the main text, along with its asymptotic theory in the appendix.
- We present empirical evidence of superiority of CBO on three representative robotics problems: long-horizon planning, double-cartpole swing-up, and demonstration free humanoid locomotion.

The paper is organized as follows. Section II reviews the zero-order methods most commonly used in robotics. Section III presents the CBO update rule and the resulting particle-search dynamics. Section IV evaluates the method on long-horizon planning, double-cartpole swing-up, and **demonstration free** humanoid locomotion, and Section V concludes. The appendix complements the main text with a unified mathematical framework for zero-order methods in Section A as an analysis tool for explaining the drawbacks of existing methods and how CBO addresses these limitations. We present a detailed CBO theoretical analysis and technical discussions in Section B, and further supporting material in Section C.

A. Notations

In this section, we summarize the notions used throughout the paper (including appendix) to improve the readability:

- r: optimization iteration index
- $x_t \in \mathcal{S} \subset \mathbb{R}^{n_s}$: state of system at time t with state space $\mathcal{S}$ of dimension n_s.
- $u_t^{(r)} = (u_{t,1}^{(r)}, \dots, u_{t,n_a}^{(r)}) \in \mathcal{U} \subset \mathbb{R}^{n_a}$: control vector at time t with n_a number of actuations, at iteration r, where $\mathcal{U} \subset \mathbb{R}^{n_a}$ is the control signal space.
- $u_{0:T-1} \in \prod_1^T \mathcal{U} \subset \mathbb{R}^{T \times n_a}$: control signal (decision vector) corresponding to planning horizon T.
- $x_{0:T}$: state trajectory driven via control trajectory $u_{0:T-1}$ through system dynamic equation starting from state x_0.
- Particle i: We use $u_{0:T-1}^{(r,i)}$ to denote the ith control signal at iteration r, which we call the ith particle.
- $\mathbb{U}^{(r)}(u_{0:T-1})$: measure (a.k.a. distribution) of control signal $u_{0:T-1}$ at iteration r.
- $\bar{u}_{0:T-1}$: the location parameter of a distribution, i.e. mean for parametric distribution or center (e.g. target consensus point) of non-parametric distribution.
- $\rho > 0$: temperature parameter controlling the selectiveness of the softmax weighting scheme.

II. Zero-Order Methods in Robotics

A. Path Integral Methods

Path integral (PI) methods introduced in [12] was derived from an information-theoretic perspective with stochastic process theory. Subsequently, a receding-horizon control variant, known as MPPI, was proposed in [13].

The original derivation of PI updates happen in the state space. In this paper, we are mostly interested in the control space solution directly. One can show that the control update can be computed as an expectation over sampled control trajectories weighted by their costs. Given a current guess of the solution $\bar{u}_{0:T-1}^{(r)}$, the algorithm samples N random controls

$$u_{0:T-1}^{(r,i)} \sim \mathcal{N}(\bar{u}_{0:T-1}^{(r)}, \Sigma), \tag{1}$$

where i indexes the N independent samples, and Σ is a fixed covariance matrix. These random control inputs are then used to compute the next guess according to the following rule:

$$\bar{u}_{0:T-1}^{(r+1)} = \sum_{i=1}^{N} w^{(r,i)} u_{0:T-1}^{(r,i)}, \tag{2}$$

$$\text{where} \quad w^{(r,k)} = \frac{\exp(-\rho J(u_{0:T-1}^{(r,k)} \mid x_0))}{\sum_{j=1}^{N} \exp(-\rho J(u_{0:T-1}^{(r,j)} \mid x_0))}, \tag{3}$$

where $w^{(r,k)}$ is the normalized weight computed from the cost of the i^{th} trajectory, $J(\cdot \mid x_0)$ is the cost function starting from initial state x_0.

B. Covariance Matrix Adaptation

Path integral methods use constant covariance matrix Σ, which caused issues in higher dimension (see detailed discussion in Remark 7 in appendix). The solution is to adapt the covariance matrix so that the sampled population can concentrate on important directions centered at the current solution, enhancing a more refined search. This approach is known as Covariance Matrix Adaptation (CMA), on top of which, CMA-ES introduced a widely used evolution path [8].

As a **simplified** introduction, CMA-ES maintains a Gaussian search distribution over the control sequences and adapts its mean, covariance, and step size α based on ranked samples [8]. At iteration r, offspring are sampled according to (4), then evaluated through the cost function J:

$$u_{0:T-1}^{(r,i)} = \bar{u}_{0:T-1}^{(r)} + \alpha^{(r)} A^{(r)} \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, I), \tag{4}$$

where A is the decomposition of the covariance matrix Σ as stated below:

$$\Sigma^{(r)} = (\alpha^{(r)})^2 A^{(r)} (A^{(r)})^\top. \tag{5}$$

The mean is then updated only using an elite proportion of the current population and a moving average trick to stabilize the updates, which we formulate in (6). The covariance is adapted to capture successful search directions as (7), using $y^{(r,i)}$ in (8), which is the deviation of particle i from the current population mean.

$$\bar{u}_{0:T-1}^{(r+1)} = (1-\alpha^{(r)}) \bar{u}_{0:T-1}^{(r)} + \alpha^{(r)} \sum_{i=1}^{N_e} w^{(r,i)} u_{0:T-1}^{(r,i)} \tag{6}$$

$$\Sigma^{(r+1)} = (1-\alpha^{(r)}) \Sigma^{(r)} + \alpha^{(r)} \sum_{i=1}^{N_e} w^{(r,i)} y^{(r,i)} y^{(r,i)\top} \tag{7}$$

$$y^{(r,i)} = u_{0:T-1}^{(r,i)} - \bar{u}_{0:T-1}^{(r)}, \tag{8}$$

where N_e is the number of elite particles. One choice of pertaining $w^{(r,i)}$, the weight for particle i, can be done via calculating (3). The step size $\alpha^{(r)}$ is adjusted by a separate evolution path to control the overall search scale. This mechanism allows CMA-ES to adaptively stretch or shrink the sampling distribution along promising directions.

C. Discussion

In the appendix, we cast the popular zero-order optimization methods used for trajectory optimization in robotics within a general mathematical framework (see (16) and theoretical discussions in Appendix A). This perspective provides insight into their limitations and helps explain why our proposed method in the next section addresses them.

III. CBO Method Overview

To address the challenges of parameterized sampling-based optimization (see detailed analyis in Section A in appendix), we introduce consensus-based optimization (CBO) as a population-based method. Instead of explicitly parameterizing a search distribution at iteration r, CBO represents it with a population of particles $\{u_{0:H}^{(r,i)}\}_{i=1}^{N}$. Under mild assumptions, CBO is known to converge to the global optimum [10, 11].

In continuous time, each particle follows the stochastic differential equation with W denoting Brownian motion.

$$du_{0:T}^{(r,i)} = -\lambda \left(u_{0:T}^{(r,i)} - \bar{u}_{0:T}^{(r)} \right) dr + \sigma \left| u_{0:T}^{(r,i)} - \bar{u}_{0:T}^{(r)} \right| dW^{(r,i)}, \tag{9}$$

where r denotes the iteration index, λ is the decay rate toward the consensus point, and σ controls the exploration intensity.

The consensus point is the cost-weighted average of the particle population,

$$\bar{u}_{0:T}^{(r)} = \sum_{i} w^{(r,i)} u_{0:T}^{(r,i)}, \tag{10}$$

where the weights (see Equation (3)) favor low-cost rollouts. In discrete time approximation, (9) becomes

$$\Delta u^{(r,i)} = u_{0:T-1}^{(r+1,i)} - u_{0:T-1}^{(r,i)} = -\lambda \left(u_{0:T-1}^{(r,i)} - \bar{u}_{0:T}^{(r)} \right) \Delta r + \sigma \sqrt{\Delta r} \left| u_{0:T-1}^{(r,i)} - \bar{u}_{0:T}^{(r)} \right| \Delta W^{(r,i)}, \tag{11}$$

where $\Delta W^{(r,i)} \sim \mathcal{N}(0, I)$.

The term $\left| u_{0:T-1}^{(r,i)} - \bar{u}_{0:T}^{(r)} \right|$ in Equation (11) makes particles that are far from the current consensus explore more

Algorithm 1: CBO for trajectory optimization

Input: Initial state $x_0 \in \mathcal{S}$, SDE integration time Δr, initial population $\{u_{0:T-1}^{(0,i)}\}$ of size N, parameters σ, λ

while *stopping criterion is not met* **do**
  for $i = 1, \ldots, N$ **do**
    Evaluate the rollout cost of particle i
  Calculate the target consensus point $\bar{u}_{0:T}$ with (10)
  for $i = 1, \ldots, N$ **do**
    Update the particle using the drift and exploration terms in (11)

Output: final population $\{u_{0:T-1}^{(r^*,i)}\}$ at iteration r^*

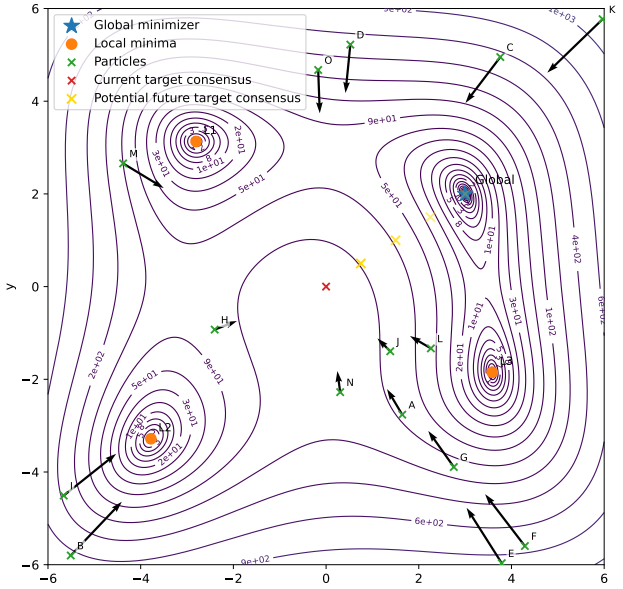


Fig. 1: Illustration of how CBO can keep exploring on a nonconvex objective with one global optimizer and several local optimizers. The particle drift is toward the consensus point rather than directly toward a local gradient direction.

aggressively, while particles near promising regions contract automatically.

Figure 1 gives an intuitive view of this behavior on a stylized nonconvex objective. Even when the current consensus is not yet at the global optimizer, particles are not constrained to make only local refinements around a single Gaussian center. Different particles can approach the consensus from different directions, and once some of them discover a better basin, the consensus shifts and redirects the rest of the population.

Algorithmically, each iteration consists of three simple steps: evaluate the rollout cost of every particle, compute the weighted consensus point, and update the particle population according to (11). See Algorithm 1. These operations parallelize naturally on modern simulators and accelerators. The appendix offers additional discussions about advantages of CBO, the supporting (see Section B-B) and technical discussion including computational cost in Section B-C and choice of hyperparameters in Section B-D.

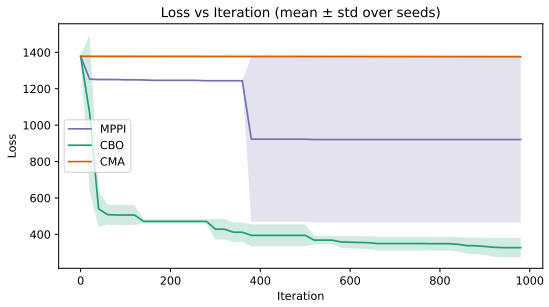


Fig. 2: Long-horizon planning benchmark. CBO reaches substantially lower best-population loss than MPPI and CMA on this deceptive planning task.

IV. Experiments

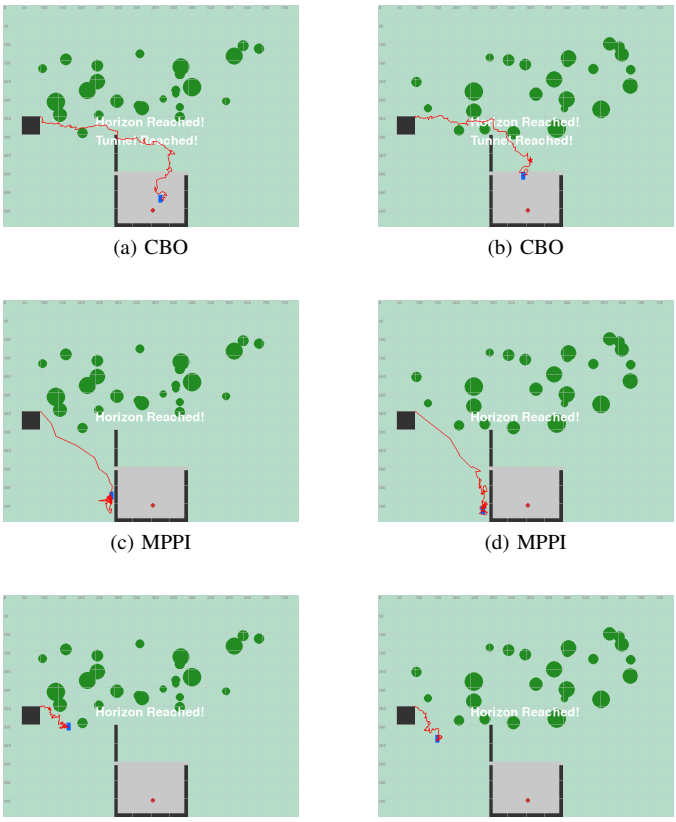
We evaluate CBO on three representative robotics problems chosen to stress different failure modes of local zero-order optimization: long-horizon planning, double-cartpole swing-up, and demonstration free humanoid control. Across all comparisons, methods share the same simulation environment, initial particle population whenever possible, and aligned optimization hyperparameters to keep the comparison focused on search behavior rather than tuning. Our implementation of the algorithms and experiments can be found in https://github.com/Atarilab/cbo_to.

A. Long-Horizon Planning

Our first example is a deliberately difficult planning task with a low-dimensional state but a horizon of $T = 100$, resulting in a 200-dimensional control decision variable. A point-mass agent must navigate through a constrained environment with obstacles and a narrow tunnel. The setup is designed so that directly approaching the left wall corresponds to a poor local minimum, while the globally favorable behavior requires a longer and less obvious route through the tunnel. One important note is that we created a tricky scenario where the green disk in Figure 3 are soft obstacles (the agent can travel through without affecting dynamic but only induces a high cost), while the wall can not be penetrated. For detailed experiment setup and result analysis, see Section D-A in the appendix, where we repeated some plots already existed in the main text for ease of reading.

This example is useful because it separates optimization difficulty from model complexity: the dynamics are simple, but the search landscape is highly deceptive. In this regime, local sampling methods often spend their budget refining trajectories that remain trapped near the visible but suboptimal basin. As shown in Figure 2, CBO performs better because different particles can keep exploring distinct routes while the consensus shifts toward lower-cost regions.

The key point is not only that CBO achieves lower loss, but that it does so on a problem where the globally useful route is geometrically narrow. This is exactly the regime where a locally adapted sampling distribution can become overconfident too early. The particle-wise exploration of CBO gives the



(a) CBO (b) CBO

(c) MPPI (d) MPPI

(e) CMA (f) CMA

Fig. 3: Qualitative long-horizon planning comparison across two obstacle layouts. CBO consistently finds routes through the tunnel, while MPPI and CMA remain trapped in poor local solutions.

optimizer more chances to discover trajectories that first move away from the apparent short-term descent direction and only later recover lower total cost. The qualitative trajectories in Figure 3 make this difference visually clear.

B. Double Cartpole

The second task is a double-cartpole swing-up problem with a long horizon and deliberately restrictive control limits. The cart must coordinate two coupled poles under difficult dynamics, making the objective highly nonconvex even though the state dimension is moderate. The system is illustrated in Figure 11 in the appendix, with detailed experimental setup and experimental results analysis in Section D-B in appendix (where we repeated some plots already existed in the main text for ease of reading). In our experiments, CBO achieves the best optimization performance as shown in Figure 4.

C. Demonstration Free Humanoid Locomotion

Our final example is demonstration-free locomotion for the 23-DoF Unitree G1 humanoid. The algorithm searches over a sequence of PD targets to move the robot base toward a desired terminal configuration within a short horizon **without relying on any demonstrated trajectory**. Even in this comparatively high-dimensional setting, CBO remains competitive because it

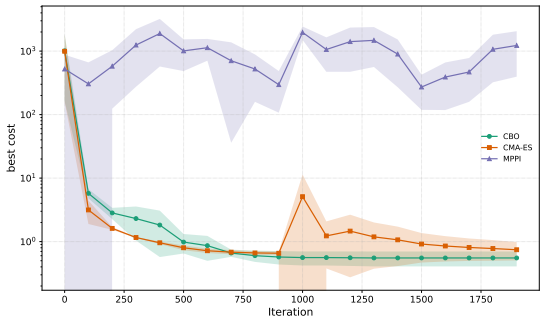


Fig. 4: Benchmark results on the double-cartpole problem. CBO achieves the best loss with relatively small variance over runs.

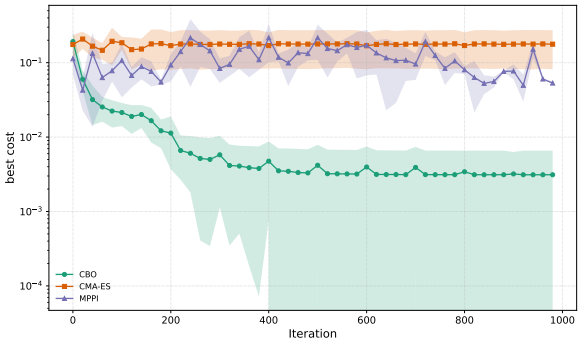


Fig. 5: Humanoid benchmark with population size $N = 10{,}000$. CBO outperforms MPPI and CMA-ES by a large margin in best-population loss.

can retain population diversity without committing too early to a single local Gaussian approximation. See the benchmark curve in Figure 5 and a set of optimized motion snapshots in Figure 6.

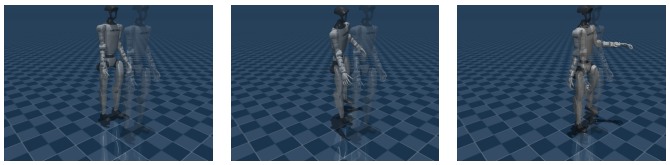


Fig. 6: Snapshots of a CBO-optimized G1 locomotion trajectory. The sequence shows the robot moving from the initial standing pose toward the target base configuration.

V. Conclusion and Acknowledgement

We introduced CBO to the robotics community as a step toward global trajectory optimization. Through three challenging trajectory optimization problems we demonstrated that CBO can find solutions with significantly lower cost for the tasks considered. In addition, by presenting a general mathematical framework for global optimization, we analyzed the limitations of widely used zero-order optimization methods in robotics and explained how CBO addresses these shortcomings. This work was partially supported by the Huawei-TUM joint laboratory.

References

[1] Xudong Sun, Armand Jordana, Massimo Fornasier, Jalal Etesami, and Majid Khadiv. Consensus-based optimization (cbo): Towards global optimality in robotics, 2026.

[2] MJX. Mujoco3, 2023.

[3] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

[4] Armand Jordana, Jianghan Zhang, Joseph Amigo, and Ludovic Righetti. An introduction to zero-order optimization techniques for robotics. *arXiv preprint arXiv:2506.22087*, 2025.

[5] Chaoyi Pan, Zeji Yi, Guanya Shi, and Guannan Qu. Sampling-based methods for optimal control: Theory, algorithms, and applications. In *2025 IEEE 64th Conference on Decision and Control (CDC)*, pages 3775–3793. IEEE, 2025.

[6] Grady Williams, Paul Drews, Brian Goldfain, James M Rehg, and Evangelos A Theodorou. Aggressive driving with model predictive path integral control. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 1433–1440. IEEE, 2016.

[7] Reuven Y Rubinstein and Dirk P Kroese. *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer Science & Business Media, 2004.

[8] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2):159–195, 2001.

[9] René Pinnau, Claudia Totzeck, Oliver Tse, and Stephan Martin. A consensus-based model for global optimization and its mean-field limit. *Mathematical Models and Methods in Applied Sciences*, 27(01):183–204, 2017.

[10] José A Carrillo, Young-Pil Choi, Claudia Totzeck, and Oliver Tse. An analytical framework for consensus-based global optimization method. *Mathematical Models and Methods in Applied Sciences*, 28(06):1037–1066, 2018.

[11] Massimo Fornasier, Timo Klock, and Konstantin Riedl. Consensus-based optimization methods converge globally. *SIAM Journal on Optimization*, 34(3):2973–3004, 2024.

[12] Evangelos Theodorou, Jonas Buchli, and Stefan Schaal. A generalized path integral control approach to reinforcement learning. *The Journal of Machine Learning Research*, 11:3137–3181, 2010.

[13] Grady Williams, Andrew Aldrich, and Evangelos A Theodorou. Model predictive path integral control: From theory to parallel computation. *Journal of Guidance, Control, and Dynamics*, 40(2):344–357, 2017.

[14] Patrick M Wensing, Michael Posa, Yue Hu, Adrien Escande, Nicolas Mansard, and Andrea Del Prete. Optimization-based control for dynamic legged robots. *IEEE Transactions on Robotics*, 40:43–63, 2023.

[15] Sebastien Labbe and Andrea Del Prete. Analytical integral global optimization. In *Proceedings of the 7th Annual Learning for Dynamics & Control Conference*, volume 283 of *Proceedings of Machine Learning Research*, pages 711–722. PMLR, 04–06 Jun 2025.

[16] Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, April 2017.

[17] Takato Miura, Naoki Akai, Kohei Honda, and Susumu Hara. Spline-interpolated model predictive path integral control with stein variational inference for reactive navigation. (arXiv:2404.10395), April 2024. arXiv:2404.10395 [cs].

[18] Robert Tjarko Lange. evosax: Jax-based evolution strategies. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*, pages 659–662, 2023.

APPENDIX A
PRELIMINARIES ON A THEORETICAL ANALYSIS FRAMEWORK FOR ZERO-ORDER OPTIMIZATION METHOD

A. The big picture of global optimization

Trajectory optimization finds a control sequence $u_{0:T-1}$ that minimizes a cost function J given an initial state x_0:

$$\min_{u_{0:T-1}} \quad J(u_{0:T-1} \mid x_0), \tag{12}$$

where the cost is defined to be stagewise while satisfying the system's dynamics $x_{t+1} = dyn(x_t, u_t)$:

$$J(u_{0:T-1} \mid x_0) = \sum_{t=0}^{T-1} \ell_t(x_t, u_t) + \ell_T(x_T) \tag{13}$$

$$\text{s.t.} \quad x_0 = x_{init}, \;\; x_{t+1} = dyn(x_t, u_t), \tag{14}$$

where $\{\ell_t\}$ and $\ell_T(x_T)$ denote the running and terminal costs, respectively.

Achieving global optimality in trajectory optimization requires overcoming the limitations of traditional gradient-based optimization techniques that often get stuck in local minima of the objective function $J(u_{0:T-1}|x_0)$ [14]. One approach to mitigating this issue is to smooth the objective landscape by integrating the cost over a neighborhood of the current guess of the solution [15].

Formally, this smoothing can be defined given a probability distribution, $\mathbb{U}$, over the controls. More precisely, the smoothing surrogate of the objective function can be written as:

$$\xi(J \mid \mathbb{U}) = \int J(u_{0:T-1}|x_0) d\mathbb{U}(u_{0:T-1}). \tag{15}$$

Remark 1. *This smoothing can be interpreted as applying a low-pass filter to the cost landscape with a given bandwidth. A heavily smoothing filter may remove many local optima around the current solution and thus enable better solutions. However, it may also shift the global optimum of the surrogate landscape away from that of the original objective. Consequently, it is natural to begin the optimization with a strongly smoothing filter and gradually reduce its magnitude.*

Now, one may search for the probability distribution $\mathbb{U}$ that minimizes $\xi(J \mid \mathbb{U})$. This naturally leads to an iterative optimization procedure in the spirit of gradient descent.

$$\mathbb{U}^{(r+1)} = \mathbb{U}^{(r)} - \alpha \Delta \xi(J \mid \mathbb{U}^{(r)}), \tag{16}$$

where we use Δ to denote the variation operation to the distribution.

Remark 2. *Typically, the distribution of controls $\mathbb{U}$ is chosen to be a multivariate Gaussian distribution. This is often referred to as randomized smoothing [16]. In this setting, we can write $\mathbb{U}(\cdot \mid \bar{u}_{0:T-1}) = \mathcal{N}(\bar{u}_{0:T-1}, \Sigma)$, where $\bar{u}_{0:T-1}$ is the mean of the gaussian distribution and Σ is a fixed covariance matrix. In this case, the iterative procedure can be expressed as an iteration over the mean of the trajectory:*

$$\bar{u}_{0:T-1}^{(r+1)} = \bar{u}_{0:T-1}^{(r)} - \alpha \nabla \xi(J \mid \mathbb{U}(\cdot \mid \bar{u}_{0:T-1}^{(r)})) \tag{17}$$

Note that the gradient is taken with respect to $\bar{u}_{0:T-1}$. The control mean can then be viewed as an estimate for the solution of the original minimization problem defined in (12)*.*

Remark 3. *To enable global optimization, the following requirements need to be satisfied for representing $\mathbb{U}$:*

- *For high-dimensional decision variables, the probability distribution $\mathbb{U}$ should be able to concentrate on the "important" regions of the solution space with favorable objective values and facilitate finite sample size approximation.*
- *The probability distribution $\mathbb{U}$ should be able to adapt and shrink its support to enable a refined search for the global optima.*

B. Gaussian Smoothing Interpretation of PI Updates

The path integral update can be interpreted as a form of Gaussian smoothing over the objective function J [4]. To be more precise, one can define a surrogate of the form

$$\xi(J \mid \mathbb{U}) = -\frac{1}{\rho} \log \left(\int e^{-\rho J(u_{0:T-1}|x_0)} d\mathbb{U}(u_{0:T-1}) \right). \tag{18}$$

Then, if we consider $\mathbb{U}(\cdot \mid \bar{u}_{0:T-1}) \sim \mathcal{N}(\bar{u}_{0:T-1}, \Sigma)$, the path integral updates in (2) can be recovered by applying (16), with the gradient taken with respect to the mean $\bar{u}_{0:T-1}$. The following proposition formalizes this result.

Proposition 1. *The update for the mean of distribution $\bar{u}_{0:T-1}$ under the PI framework in* (2) *can be written as*

$$\bar{u}_{0:T-1}^{(r+1)} = \bar{u}_{0:T-1}^{(r)} - \gamma\Delta(\bar{u}_{0:T-1}), \tag{19}$$

where $\Delta(\bar{u}_{0:T-1})$ is the solution to the following constrained optimization problem with γ being the Lagrange multiplier.

$$\arg\min_{\Delta(\bar{u}_{0:T-1})} \xi(J \mid \mathbb{U}(\cdot \mid \bar{u}_{0:T-1} + \Delta(\bar{u}_{0:T-1}))) \tag{20}$$

$$s.t.\ KL\Big(\mathbb{U}(u \mid \bar{u}_{0:T-1} + \Delta(\bar{u}_{0:T-1}))|\mathbb{U}(u \mid \bar{u}_{0:T-1})\Big) = \beta, \tag{21}$$

where β defines the Kullback–Leibler (KL) divergence metric between the current distribution and new distribution.

We invite readers who are interested in the details to find the proof in Section C-A in the appendix.

C. Further Remarks on Zero-Order Methods in Robotics

Remark 4. *The Lagrange multiplier $\gamma = \frac{d\xi(J|\mathbb{U};\bar{u}_{0:T-1}+\Delta;\beta)}{d\beta}$ decides how much the optimal surrogate function will change per β change. In practice, this constrained optimization problem is not solved explicitly; instead, the update in* (2) *is applied directly, in which case the parameter β could vary per iterations.*

Remark 5 (Particle dynamic under parameterized distribution updates)**.** *Consider two arbitrarily chosen particles (random control sequences) from iterations r and $r+1$, the difference between them can be written as*

$$\begin{aligned} u_{0:T-1}^{(r,i)} - u_{0:T-1}^{(r+1,j)} &= \bar{u}_{0:T-1}^{(r)} + \Sigma^{\frac{1}{2}}\epsilon^{(r,i)} - \bar{u}_{0:T-1}^{(r+1)} - \Sigma^{\frac{1}{2}}\epsilon^{(r+1,j)} \\ &= \Sigma^{\frac{1}{2}}(\epsilon^{(r,i)} - \epsilon^{(r+1,j)}) + \frac{1}{\gamma}\Delta(\bar{u}_{0:T-1}), \end{aligned} \tag{22}$$

where $\{\epsilon^{(r,i)}\}$ are i.i.d. standard normal variables. Thus, the potential improvements of $u^{(r+1,i)}$ over $u^{(r,j)}$ come from the major effect of $\Delta(\bar{u}_{0:T-1})$ plus a random exploration term $\Sigma^{\frac{1}{2}}(\epsilon^{(r,i)} - \epsilon^{(r+1,j)})$. Since the particles are reset after each iteration, there is no distinction between them (particles are "forgotten" after each iteration), resulting in homogeneous exploration. Any potential improvements beyond $\Delta(\bar{u}_{0:T-1})$ comes from repeatedly exploring the same mechanism of random directions.

Remark 6. *Note that when ρ is big enough in* (18)*:*

$$\xi(J \mid \mathbb{U}(\cdot \mid \bar{u}_{0:T-1})) \approx \inf J(u_{0:T-1} \mid x_0). \tag{23}$$

Thus, optimizing the surrogate is approximately equivalent to optimizing the lower bound, while the constraint penalizes large changes by restricting how much the distribution can shift at each iteration. With a finite sample size, this involves choosing the best performing $u_{0:T-1}$ among the sampled population.

Remark 7 (Curse of dimensionality in finite sample approximation of Gaussian expectation)**.** *When the decision variable lies in an extremely high-dimensional space, a finite sample size cannot realistically approximate the distribution; thus, the lower bound of a finite sample does not approximate the mathematical formulation of* (18) *faithfully.*

Remark 8 (Connection between CMA and CEM)**.** *Setting $w^{(r,i)} = \frac{1}{N_e}$ with elite particles $N_e < N$ in* (7) *gives the Cross Entropy Method (CEM) [7].*

Remark 9 (curse of distribution parametrization)**.** *When a parameterized distribution is employed, its shape can be adjusted only through a limited number of parameters, which constrains the expressive flexibility of $\mathbb{U}(u_{0:T-1})$. The Gaussian distribution has additional innate geometric structures, such as symmetry, which typically do not fit control signal distributions. For instance, flipping the control signal with respect to an arbitrary center might result in a control signal that leads to undesirable behavior. In Figure 7, we illustrate the difference between a Gaussian distribution and an irregular, non-parametric distribution. The irregular distribution can develop longer, potentially asymmetric tails along important directions, whereas a shrinking Gaussian is much less flexible in accommodating such behavior. In Section B, we introduce CBO and explain how an irregular, non-parameteric distribution can be achieved, which also has the ability to concentrate on important regions.*

APPENDIX B
FURTHER DISCUSSIONS ON CBO

A. Advantages of CBO in global optimization

The main text already provides an intuitive illustration of why CBO can better handle nonconvex objectives with multiple local minima. Here we only record the key technical takeaway for the appendix: CBO updates are driven by particle motion toward a consensus point together with particle-dependent exploration, which allows the empirical search distribution to remain adaptive and non-Gaussian.

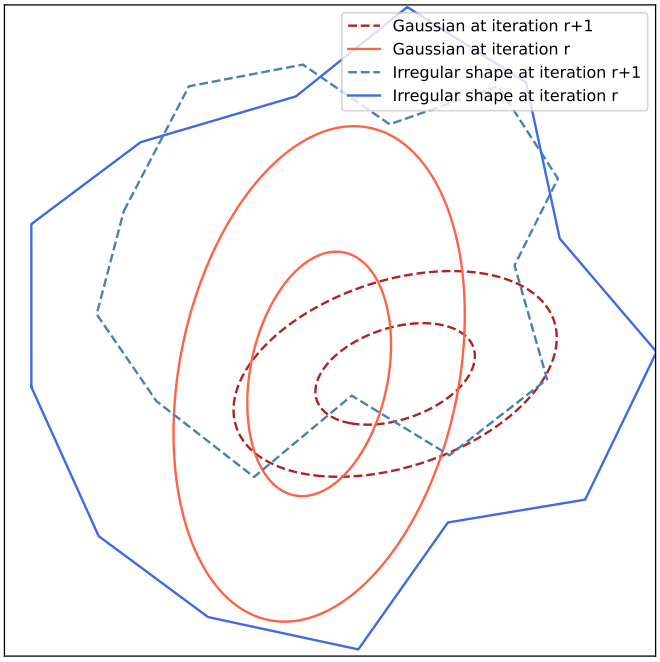


Fig. 7: An irregular, non-parametric distribution can focus probability on important directions by having longer tails, whereas a shrinking Gaussian is constrained by its fixed symmetric shape and has less flexibility.

- The dynamic of each particle is not directly driven by the gradient of the objective function, but by a drift term that pulls the particle toward the target consensus point at the current iteration in (9). As a result, the dynamics are less sensitive to local objective traps than purely local descent mechanisms.
- The inertia of each particle enables local exploration around the previous particle position. In contrast, local sampling-based methods typically discard individual particle histories after fitting a new proposal distribution.
- As particles move toward the target consensus point, they explore along different directions. Once a better region is found, the consensus point shifts and redirects the full population, yielding a more diverse exploration mechanism than the homogeneous exploration used in PI-style updates; see (22).
- The non-parametric nature of CBO allows it to adapt to complex cost landscapes more effectively than parameterized distributions, which may struggle to capture intricate solution spaces. The particles gradually gather in the vicinity of the target consensus point, analogous to the covariance matrix shrinkage in CMA. Unlike CMA, however, the empirical distribution of CBO is not Gaussian-shaped, but instead exhibits an irregular shape as illustrated in Figure 7.

Remark 10 (How CBO alleviates the curse of dimensionality and the curse of distribution parameterization)**.** *In Remark 7, we pointed out that when the decision variables have high dimensions, faithfully approximating the cost surrogate in* (18) *becomes unrealistic. CMA trys to resolve this issue by adapting its covariance matrix to let finite samples better concentrate on favorable directions, with potential shrunken eigenvalues, but is still restricted by the parameterized form of the as we noted in Remark 9. Instead, the population for CBO is evolving via the dynamic* (9)*, which offers a more flexible adaptive capacity to capture interesting regions of the decision variable space. In Section B-B, we demonstrate such an irregular, non-parametric distribution of solutions from CBO is effective in achieving global optimality.*

B. Properties of CBO

First we show CBO falls into the global optimization framework of (16) with the surrogate function in (15) under the assumption that $J(u \mid x_0)$ satisfies the so-called inverse continuity property and relaxed Lipschitz continuous property, which we state below:

$$\frac{1}{L(u)}\|J(u \mid x_0) - J(u^* \mid x_0)\| \leq \|u - u^*\| \tag{24}$$

$$\leq \frac{1}{\eta}\|J(u \mid x_0) - J(u^* \mid x_0)\| \tag{25}$$

$$\forall u \in B_\kappa(u^*) \tag{26}$$

where $B_\kappa(u^*)$ is a neighborhood of the optima u^* with radius κ, and $\eta > 0$ is a constant.

Formally, to demonstrate the irregular, non-parametric distribution of particles of CBO is effective in achieving global optimality, we have the following results:

Proposition 2. *Without loss of generality, assume* $J(u^* \mid x_0) = 0$. *When Equations* (24) *to* (26) *hold, there exist* $\rho, r^* > 0$ *large enough, where*

$$r^* \propto \frac{1}{2\lambda - n_a \times T\sigma^2}, \tag{27}$$

such that, with high probability, CBO improves the surrogate function following the formulation in (16) *after at most* r^* *iterations, i.e., we have*

$$\begin{aligned} &\int J(u_{0:T-1}|x_0) d\mathbb{U}^{(r^*)}(u_{0:T-1} \mid \bar{u}_{0:T-1}^{(r^*)}) \\ <&\int J(u_{0:T-1}|x_0) d\mathbb{U}^{(0)}(u_{0:T-1} \mid \bar{u}_{0:T-1}^{(0)}). \end{aligned} \tag{28}$$

In addition, due to the structure of $\mathbb{U}$ *evolution driven by the CBO dynamic in* (9)*, the target consensus point converges to the global optimizer with high probability.*

Proof is in Section C-B of the supplementary material.

Remark 11 (choice of decay rate λ)**.** *Equation* (27) *gives us a convenient reference of selection of decay rate* λ *with respect to the noise level* σ^2 *to ensure convergence of the dynamics of* (11)*. Concretely, the convergence can be ensured by* $2\lambda > n_a \times T \times \sigma^2$*. For hyperparameter tuning, we recommend simply calculate the lowest* λ *that satisfies this formula and then try a grid of values, while ensuring* $\lambda \Delta r < 1$*. Increasing* λ *improves convergence rate but too large* λ *breaks the performance, see an empirical comparision of different* λ *in Figure 8 in the appendix to understand this tradeoff.*

C. Computational cost (solve times)

The bottleneck of the computation time is the **simulation rollout time**. The statistics in Table I are generated on a NVIDIA GeForce RTX 5090 Laptop GPU with a 32 × AMD Ryzen 9 9955HX3D 16-Core processor running a Linux operating system (6.17.0-109014-tuxedo), for the double-cartpole experiment with $N = 1000$ particles, where **jax compilation time** and **per iteration computation time** are reported: The table shows no significant difference between different zero-order methods in per-iteration computation time.

TABLE I: Runtime comparison across algorithms.

Algorithm	Compile (ms)	Mean (ms)	Std (ms)
CBO	6638.30	336.43	1.02
CMA-ES	11369.38	351.28	12.53
MPPI	6548.47	372.48	35.96

D. More discussions on drift/decay rate λ

We mentioned in Remark 11 that λ (drift/decay) should be chosen to match the diffusion strength σ (for which we fix the value for fair comparison with competitor algorithms) to ensure convergence of the dynamics as presented in our theoretical results. Concretely, the convergence can be ensured by $2\lambda > n_a \times T \times \sigma^2$. In practice, we simply calculate the lowest λ that satisfies this formula and then try a few different integer values, and ensure $\lambda \Delta r < 1$. Increasing λ improves convergence rate but too large λ breaks the performance, see the comparision of different λ in Figure 8.

APPENDIX C
PROOFS AND ADDITIONAL APPENDIX MATERIAL

To ensure consistent cross-referencing with the main text, the equation numbering is continued in this appendix.

A. Proof of Proposition 1.

We first prove the following helper lemma.

Lemma 1. *Let* $x \in \mathbb{R}^d$ *be distributed as*

$$x \sim \mathcal{N}(\mu, \Sigma),$$

where $\mu \in \mathbb{R}^d$ *is the mean and* $\Sigma \in \mathbb{R}^{d \times d}$ *is the covariance matrix. Then,*

$$\mathcal{I}(\mu) = \Sigma^{-1}.$$

Proof. The log-likelihood for a single observation is

$$\ell(\mu) = \log p(x \mid \mu) = -\frac{1}{2}(x - \mu)^\top \Sigma^{-1}(x - \mu) + C.$$

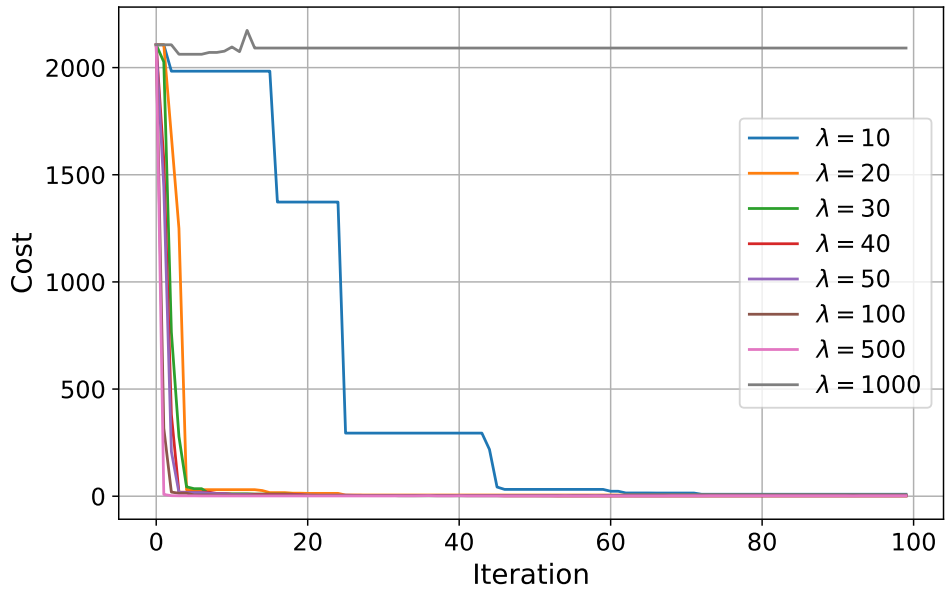


Fig. 8: Too large λ hurts performance

The score function is

$$\nabla_\mu \ell(\mu) = \Sigma^{-1}(x - \mu).$$

The Fisher Information Matrix is defined as

$$\mathcal{I}(\mu) = \mathbb{E}\Big[\nabla_\mu \ell(\mu)\, \nabla_\mu \ell(\mu)^\top\Big].$$

Using $\mathbb{E}[(x-\mu)(x-\mu)^\top] = \Sigma$, we obtain $\mathcal{I}(\mu) = \Sigma^{-1}\,\Sigma\,\Sigma^{-1} = \Sigma^{-1}$. For n independent and identically distributed samples, the Fisher information becomes $\mathcal{I}_n(\mu) = n\,\Sigma^{-1}$. □

We are now ready to prove Proposition 1.

Proof. The solution to the constrained optimization problem in (20) with constraint (21) can be derived using Lagrange multipliers, leading to an update rule for $\bar{u}_{0:T-1}$ that incorporates both the gradient of the surrogate cost and the KL divergence constraint, i.e.,

$$\mathcal{L}(\Delta(\bar{u}_{0:T-1})) := \xi(J \mid \mathbb{U}(\cdot \mid \bar{u}_{0:T-1} + \Delta(\bar{u}_{0:T-1}))) + \gamma\Big(KL(\mathbb{U}(u \mid \bar{u}_{0:T-1} + \Delta(\bar{u}_{0:T-1}))|\mathbb{U}(u \mid \bar{u}_{0:T-1})) - \beta\Big). \tag{29}$$

When $\mathbb{U}(u \mid \bar{u}_{0:T-1})$ is Gaussian, the covariance matrix remains the same, we have

$$KL(\mathbb{U}(u \mid \bar{u}_{0:T-1} + \Delta(\bar{u}_{0:T-1}))|\mathbb{U}(u \mid \bar{u}_{0:T-1})) = \frac{1}{2}\|\Delta(\bar{u}_{0:T-1})\|^2_{\mathcal{I}_{\bar{u}_{0:T-1}}} = \beta, \tag{30}$$

where $\mathcal{I}_{\bar{u}_{0:T-1}} = \Sigma^{-1}$ is the Fisher information matrix of $\mathbb{U}(u_{0:T-1})$ with respect to its mean (see Lemma 1).

This result can be derived from the closed-form expression of the KL-divergence between two multivariate Gaussian distributions below with $\Sigma_0 = \Sigma_1 = \Sigma$:

$$\mathrm{KL}(\mathcal{N}(\mu_0, \Sigma_0) \parallel \mathcal{N}(\mu_1, \Sigma_1)) = \frac{1}{2}\left[\log\frac{|\Sigma_1|}{|\Sigma_0|} - \dim(\mu) + \mathrm{tr}\big(\Sigma_1^{-1}\Sigma_0\big) + (\mu_1 - \mu_0)^\top \Sigma_1^{-1}(\mu_1 - \mu_0)\right] \tag{31}$$

With (30), the gradient of the constraint is $\mathcal{I}_{\bar{u}_{0:T-1}}\Delta(\bar{u}_{0:T-1})$, leading to the following stationary condition of the Lagrangian:

$$\nabla_{\Delta(\bar{u}_{0:T-1})}\xi(J \mid \mathbb{U}(\cdot \mid \bar{u}_{0:T-1} + \Delta(\bar{u}_{0:T-1})))\big|_{\Delta(\bar{u}_{0:T-1})=\Delta(\bar{u}_{0:T-1})^*} + \gamma\Sigma^{-1}\Delta(\bar{u}_{0:T-1})^* = 0, \tag{32}$$

where we use superscript $*$ to denote the optimal solution.

Below, we compute the first term in the above equation. Recall that

$$\xi(J \mid \mathbb{U}) = -\frac{1}{\rho}\log\left(\int e^{-\rho J(u_{0:T-1}|x_0)} d\mathbb{U}(u_{0:T-1})\right) = -\frac{1}{\rho}\log\Big(\mathbb{E}_{u\sim\mathbb{U}}[\exp\left(-\rho J(u|x_0)\right)]\Big). \tag{33}$$

Therefore, we get

$$\nabla_{\Delta(\bar{u}_{0:T-1})}\xi(J \mid \mathbb{U}(\cdot \mid \bar{u}_{0:T-1} + \Delta(\bar{u}_{0:T-1}))) = -\frac{1}{\rho}\frac{\nabla_{\Delta(\bar{u}_{0:T-1})}\mathbb{E}_{u\sim\mathbb{P}}[\exp\left(-\rho J(u|x_0)\right)]}{\mathbb{E}_{u\sim\mathbb{P}}[\exp\left(-\rho J(u|x_0)\right)]}, \tag{34}$$

where $\mathbb{P} = \mathbb{U}(\bar{u}_{0:T-1} + \Delta(\bar{u}_{0:T-1}))$.

Recall that we use $\mathbb{U}(\cdot \mid \bar{u}_{0:T-1})$ to represent $\mathcal{N}(\bar{u}_{0:T-1}, \Sigma)$ and thus $\mathbb{P}$ represents $\mathcal{N}(\bar{u}_{0:T-1} + \Delta(\bar{u}_{0:T-1}), \Sigma)$. We can compute the gradient in the nominator of (34) using the following result.

$$\nabla_\theta \mathbb{E}_{x\sim p_\theta}[f(x)] = \nabla_\theta \int f(x)\, p_\theta(x)\, dx = \int f(x)\, p_\theta(x)\, \nabla_\theta \log p_\theta(x)\, dx \tag{35}$$

$$= \mathbb{E}_{x\sim p_\theta}[f(x)\, \nabla_\theta \log p_\theta(x)]\,. \tag{36}$$

Using the above equation and the fact that

$$\nabla_{\Delta(\bar{u}_{0:T-1})} \log d\mathbb{P} = \Sigma^{-1}(u - \bar{u}_{0:T-1} - \Delta(\bar{u}_{0:T-1})), \tag{37}$$

equation (34) becomes

$$\nabla_{\Delta(\bar{u}_{0:T-1})}\xi(J \mid \mathbb{U}(\cdot \mid \bar{u}_{0:T-1} + \Delta(\bar{u}_{0:T-1}))) = \frac{\mathbb{E}_{u\sim\mathbb{P}}[\exp\left(-\rho J(u \mid x_0)\right)\Sigma^{-1}(u - \bar{u}_{0:T-1} - \Delta(\bar{u}_{0:T-1}))]}{\mathbb{E}_{u\sim\mathbb{P}}[\exp\left(-\rho J(u|x_0)\right)]} \tag{38}$$

$$= \Sigma^{-1}\mathbb{E}_{u\sim\mathbb{U}}\Big[\frac{\exp\left(-\rho J(u + \Delta(\bar{u}_{0:T-1}) \mid x_0)\right)(u - \bar{u}_{0:T-1})}{\mathbb{E}_{u'\sim\mathbb{U}}[\exp\left(-\rho J(u' + \Delta(\bar{u}_{0:T-1})|x_0)\right)]}\Big] \tag{39}$$

$$= \Sigma^{-1}\mathbb{E}_{u\sim\mathbb{U}}[w_{\bar{u}_{0:T-1},u}(u - \bar{u}_{0:T-1})], \tag{40}$$

The equality in (39) is due to change of variable, $u \leftarrow u - \Delta(\bar{u}_{0:T-1})$.

Note that we use $\mathbb{U}(\cdot \mid \bar{u}_{0:T-1})$ to denote $\mathcal{N}(\bar{u}_{0:T-1}, \Sigma)$ and in (40),

$$w_{\bar{u}_{0:T-1},u} = \frac{\exp\left(-\rho J(u + \Delta(\bar{u}_{0:T-1}) \mid x_0)\right)}{\mathbb{E}_{u'\sim\mathbb{U}}[\exp\left(-\rho J(u' + \Delta(\bar{u}_{0:T-1})|x_0)\right)]} \approx \frac{\exp\left(-\rho J(u) \mid x_0)\right)}{\mathbb{E}_{u'\sim\mathbb{U}}[\exp\left(-\rho J(u')|x_0)\right)]}. \tag{41}$$

The above approximation is reasonable due to the KL-divergence constraint in (30) with a small enough β, i.e. the shift between the two distributions being constrained to be minimal.

Putting together (32) and (40) results in

$$\Delta(\bar{u}_{0:T-1})^* = -\frac{1}{\gamma}\mathbb{E}_{u\sim\mathbb{U}}[w_{\bar{u}_{0:T-1},u}(u - \bar{u}_{0:T-1})]. \tag{42}$$

Thus following natural gradient with rate γ, i.e., (19), we obtain the following update rule:

$$\bar{u}^{(r+1)}_{0:T-1} = \bar{u}^{(r)}_{0:T-1} - \gamma\Delta(\bar{u}_{0:T-1})^* \tag{43}$$

$$= \bar{u}^{(r)}_{0:T-1} + \mathbb{E}_{u\sim\mathbb{U}} w_{\bar{u}_{0:T-1},u}(u - \bar{u}_{0:T-1}) \tag{44}$$

$$= \mathbb{E}_{u\sim\mathbb{U}} w_{\bar{u}_{0:T-1},u} u. \tag{45}$$

The last equality comes from the fact that $\mathbb{E}_{u\sim\mathbb{U}}[w_{\bar{u}_{0:T-1},u}\bar{u}_{0:T-1}] = \bar{u}_{0:T-1}$

□

B. Proof of Proposition 2.

To prove the conclusion, we provide the following lemmas based on reformulations of what was established in [11].

Lemma 2 (Exponential decay of Lyapunv function in mean dynamic)**.** *Define* $\mathcal{V} = \int \frac{1}{2}\|u_{0:T-1} - u^*_{0:T-1}\|_2^2 d\mathbb{U}(u_{0:T-1})$ *(Lyapunov function). When Equations* (24) *to* (26) *holds, and* $u^*_{0:T-1}$ *lives in the support of* $\mathbb{U}^{(0)}$*, given* $0 < \vartheta < 1$*,* $\exists\rho(\vartheta)$ *in Equation* (3) *which is large enough, such that*

$$\mathcal{V}(\mathbb{U}^{(r)}) \leq \mathcal{V}(\mathbb{U}^{(0)})\exp\left(-r(1-\vartheta)(2\lambda - T \times n_a\sigma^2)\right) \tag{46}$$

under the mean field dynamic (see Equation(7-8) in [11]).

Proof. This is a restatement of *Theorem 3.7* of [11], □

Lemma 3 (Convergence in probability: Wasserstein distance to global optimizer)**.** *When Equations* (24) *to* (26) *hold, given* $0 < \vartheta < 1$, $\exists \rho(\vartheta)$ *in* (3) *which is large enough and iteration* $r > r^*(\underline{\mathcal{V}}) > 0$,

$$\Pr\left(\|\frac{1}{N}\sum_{1}^{N} u_{0:T-1}^{(r,i)} - u_{0:T-1}^*\|_2^2 \leq \epsilon_e\right) \geq 1 - \delta\left(\epsilon_e, N, \Delta r, n_a \times T, \underline{\mathcal{V}}\right)) \tag{47}$$

where

$$r^* = \frac{1}{1-\vartheta}\frac{1}{2\lambda - n_a \times T\sigma^2}\log\frac{\mathcal{V}(\mathbb{U}^{(0)})}{\underline{\mathcal{V}}} \tag{48}$$

and

$$\mathcal{V}(\mathbb{U}) = \frac{1}{2}\int \|\mathbb{U} - \mathbb{U}^*\|^2 d\mathbb{U}(u_{0:T-1}), \quad 0 < \underline{\mathcal{V}} < \mathcal{V}(\mathbb{U}^{(0)}). \tag{49}$$

In (47)*,* $\delta\left(\epsilon_e, N, \Delta r, n_a \times T, \underline{\mathcal{V}}\right))$ *decreases with larger population size* N *and smaller* Δr *(Euler-Maruyama time interval in* (11)*) and smaller* $\underline{\mathcal{V}}$ *(corresponding to more Euler-Maruyama integration steps in* (11)*).*

Proof. This statement is a reformulation of Theorem 3.8 (convergence of CBO under finite sample size approximation to the Fokker-Planck equation) in [11]. □

Now we present the proof of Proposition 2.

Proof. To simplify notations, we use $\mathbb{U}^{(r^*)}(u_{0:T-1})$ to replace $\mathbb{U}^{(r^*)}(u_{0:T-1} \mid \bar{u}_{0:T-1}^{(r^*)})$.

We first prove the mean-field approximation [11] case:

According to Lemma 2, under assumption Equations (24) to (26), the diffusion dynamic in (9) leads to descent of $\mathcal{V}$ exponentially.

Choose the first phase duration r_1 sufficiently large, such that $\mathcal{V}$ is small enough after the particles enter the neighborhood $B_\kappa(u_{0:T-1}^*)$ of the global optima $u_{0:T-1}^*$.

In the second phase, the particles are close enough to the global optima, i.e., they are within the neighborhood $B_\kappa(u_{0:T-1}^*)$. Once inside this neighborhood, define $L = \sup_{B_\kappa(u_{0:T-1}^*)} L(u_{0:T-1})$, which results in

$$\xi(J \mid \mathbb{U}(u_{0:T-1})) = \int \big(J(u_{0:T-1} \mid x_0) - J(u_{0:T-1}^* \mid x_0)\big) d\mathbb{U}(u_{0:T-1}) \text{ (assume } J(u_{0:T-1}^* \mid x_0) = 0) \tag{50}$$

$$\leq \int L\|u_{0:T-1} - u_{0:T-1}^*\|_2 d\mathbb{U}(u_{0:T-1}) \text{ (Lipschitz)} \tag{51}$$

$$\leq L\sqrt{\int \|u_{0:T-1} - u_{0:T-1}^*\|_2^2 d\mathbb{U}(u_{0:T-1})} \text{ (Cauchy-Schwarz inequality)} \tag{52}$$

$$= L\sqrt{2\mathcal{V}(\mathbb{U})} \tag{53}$$

Since $\mathcal{V}(\mathbb{U})$ continues to decrease exponentially under the diffusion dynamic in (9) according to Lemma 2, $\xi(J \mid \mathbb{U}(u_{0:T-1}))$ also decreases accordingly. Thus, we have

$$\int J(u_{0:T-1}|x_0)d\mathbb{U}^{(r^*)}(u_{0:T-1}) \tag{54}$$

$$\leq L\sqrt{2\mathcal{V}(\mathbb{U}^{(r^*)})} \text{ ((53))} \tag{55}$$

$$\leq L\sqrt{2\mathcal{V}(\mathbb{U}^{(0)})\exp\left(-r^*(1-\vartheta)(2\lambda - T\times n_a\sigma^2)\right)} \text{ (exponential decay of } \mathcal{V}(\mathbb{U}), Lemma\ 2) \tag{56}$$

$$= L\sqrt{2\exp\left(-r^*(1-\vartheta)(2\lambda - T\times n_a\sigma^2)\right)}\sqrt{\int \|u_{0:T-1} - u^*_{0:T-1}\|_2^2 d\mathbb{U}^{(0)}(u_{0:T-1})} \tag{57}$$

$$\leq L\frac{\sqrt{2\exp\left(-r^*(1-\vartheta)(2\lambda - T\times n_a\sigma^2)\right)}}{\eta}\sqrt{\int (J(u)-J(u^*))^2 d\mathbb{U}^{(0)}(u_{0:T-1})} \text{ (inverse continuity in (25))} \tag{58}$$

$$= L\frac{\sqrt{2\exp\left(-r^*(1-\vartheta)(2\lambda - T\times n_a\sigma^2)\right)}}{\eta}\sqrt{\int (J(u))^2 d\mathbb{U}^{(0)}(u_{0:T-1})} \text{ (using } J(u^*)=0) \tag{59}$$

$$= \frac{L\sqrt{2\exp\left(-r^*(1-\vartheta)(2\lambda - T\times n_a\sigma^2)\right)}}{\eta}\zeta(J,\mathbb{U}^{(0)})\int J(u_{0:T-1}|x_0)d\mathbb{U}^{(0)}(u_{0:T-1}) \tag{60}$$

where in the last equation we define $\zeta(J,\mathbb{U}^{(0)})$, s.t.

$$\zeta(J,\mathbb{U}^{(0)})\int J(u_{0:T-1}|x_0)d\mathbb{U}^{(0)}(u_{0:T-1}) = \sqrt{\int (J(u_{0:T-1}\mid x_0))^2 d\mathbb{U}^{(0)}(u_{0:T-1})} \tag{61}$$

Due to the exponential in (56), we can always let the iteration persist until we find a big enough r^* such that $\frac{L\sqrt{2\exp(-r^*(1-\vartheta)(2\lambda-T\times n_a\sigma^2))}}{\eta}\zeta(J,\mathbb{U}^{(0)}) < 1$, such that

$$\int J(u_{0:T-1}|x_0)d\mathbb{U}^{(r^*)}(u_{0:T-1}) < \int J(u_{0:T-1}|x_0)d\mathbb{U}^{(0)}(u_{0:T-1}) \tag{62}$$

In the finite sample approximation, a similar argument holds according to Lemma 3: Choose ϵ_e small enough in (47) such that with high probability (at the cost of large population size N, smaller Δr and long iterations), all particles reside in $B_\kappa(u^*)$. Then (62) also holds following the same arguments as in the mean-field (via choosing a even smaller ϵ_e). The target consensus point can be regarded as a convex combination of particles (see (10)): when all particles converge to the global optimizer, the convex combination converges as well.

□

Remark 12. *The conclusion in Proposition 2 is essential, as it shows that, compared to the other zero-order optimization methods considered in this paper, CBO generates a population with significantly lower average cost. While alternative methods typically produce a large number of poor-quality samples alongside a single high-performing solution, CBO yields a population whose members are, on average, well behaved and consistently low cost.*

C. Details about the illustrative objective function in Figure 1

a) Objective function.: We consider the following non-convex function:

$$f(x,y) = \underbrace{\left(x^2+y-11\right)^2 + \left(x+y^2-7\right)^2}_{h(x,y)} + \alpha\left((x-3)^2+(y-2)^2\right). \tag{63}$$

where $\alpha > 0$ is a small constant.

The first term $h(x,y)$ is the classical Himmelblau function, which is non-convex and admits a finite number of isolated minimizers. The second term is a quadratic penalty centered at $(3,2)$ that is zero at this point and strictly positive elsewhere.

b) Global minimizer.: Since $h(x,y) \geq 0$ for all (x,y) and

$$h(3,2) = 0,$$

it follows that

$$f(3,2) = 0,$$

and therefore $(3,2)$ is a global minimizer of f. Moreover, for any $(x,y) \neq (3,2)$, the penalty term satisfies

$$\alpha\left((x-3)^2+(y-2)^2\right) > 0,$$

which implies $f(x,y) > 0$. Hence, $(3,2)$ is the *unique* global minimizer of f.

c) Local minimizers.: The Himmelblau function $h(x, y)$ has four isolated minimizers located at $(3, 2)$, $(-2.805118,\ 3.131312)$, $(-3.779310,\ -3.283186)$, and $(3.584428,\ -1.848126)$, all of which achieve the same minimal value of zero for h.

After adding the quadratic penalty, the point $(3, 2)$ remains unchanged and becomes the unique global minimizer of f, while the remaining three minimizers persist as *strict local minimizers* with strictly larger objective values. Since the perturbation is smooth and α is small, these local minimizers remain isolated and lie in neighborhoods of the corresponding minimizers of h.

d) Summary.: The function $f(x, y)$ is smooth and non-convex, admits a *finite number of local minimizers*, and possesses a *single global minimizer* at $(3, 2)$. This structure makes it a convenient test problem for studying optimization dynamics in the presence of multiple local minima.

APPENDIX D
FURTHER EXPERIMENT DETAILS

A. Details on long horizon planning experiment

In this example, we study a trajectory optimization problem involving a simplified dynamical system with a low-dimensional state space but an extremely long planning horizon. As illustrated in Figure 10, the agent is modeled as a **point mass** but represented by a chassis for better visualization (shown as a small blue rectangle) that travels from an initial position (black square located at the center left) towards a goal position (marked by a small red disk) through a confined space (depicted as a large green square). The tunnel has an upward-facing opening and is bounded by a wall on the left side (shown as black slats) that extends upward, as well as a shorter wall on the right side. We deliberately design the left wall to extend upward so that reaching it corresponds to a local minimum, since the agent cannot penetrate the wall. In many robotics control problems, feasible control signals lie in very "narrow" regions. To emulate this phenomenon, we randomly place obstacles (shown as dark green disks) such that only carefully chosen control signals allow the agent to successfully navigate through the obstacles and reach the tunnel.

We use the following simple first order dynamic model without inertia (i.e., velocity can be changed instantaneously):

$$q_x(t+1) = q_x(t) + v\cos(\theta)\delta t, \tag{64}$$

$$q_y(t+1) = q_y(t) + v\sin(\theta)\delta t. \tag{65}$$

To avoid optimization in the radian space, instead of optimizing the sequence of $\theta \in [0, 2\pi], v \in \mathbb{R}$, we use $v_x = v\cos(\theta), v_y = v\sin(\theta)$ as the decision variable. The decision making horizon is set to $T = 100$ steps, and the control dimension is $n_a = 2$. Despite the simplicity of the dynamic, the number of decision variables is $n_a \times T = 200$, making it a challenging long horizon planning problem.

The cost function is composed of the following terms: the primary term is the running cost, which is the distance to the goal (red disk) at each step. On top of that, we add the control loss term, plus a penalty term for not reaching the tunnel and further penalties when confronting the obstacles along the planned trajectories. In summary, the cost function is defined as:

$$\begin{aligned} J = &\sum_{t=0}^{T-1} \left(\|q(t) - q_{goal}\|^2 + \gamma_v \|v(t)\|^2 \right) \\ &+ I_{\text{not in tunnel}}\left(q(T-1)\right) + \sum_{i=1}^{N_{obs}} I_{\text{obstacle}}(q_{0:T-1}), \end{aligned} \tag{66}$$

where $I_{\text{not in tunnel}}$ is an indicator function that adds a large penalty if the final position is not inside the tunnel, and I_{obstacle} is an indicator function that adds a penalty if the trajectory confronts any obstacle. We set $\gamma_v = 10.0$, $I_{\text{not in tunnel}}\left(q(T-1)\right) = 1000$ if the agent is not inside the tunnel at the last step, $I_{obstacle} = 1000 \times n_c$, where n_c is the number of obstacles confronted along the trajectory. The parameters for penalty terms are chosen to ensure that circumventing the left wall and the obstacles is more cost-effective than confronting them directly, such that the optimal solution involves navigating through the tunnel and reaching the goal. Note that the velocity is not bounded but only regularized in the cost function with γ_v.

In terms of collision handling, the left wall acts as a barrier for the agent to reach the tunnel and goal (red disk). To simplify, we restrain the agent to bounce back to its original place if the current velocity would bring the agent into contact with the wall. On the other hand, when confronting the disk barrier, we simply add a penalty to the cost function but do not restrain the dynamic accordingly (i.e., the agent can penetrate the disk obstacle but incurs a large cost). The agent needs to make long term planning to avoid getting stuck in local minima, especially when the obstacles are placed closely.

For each repetition of the experiment, we create the same environment for competing algorithms by randomly sampling obstacles. We compare CBO with MPPI and CMA, where we implemented CMA according to (6) and (7). We set population size $N = 1000$. The covariance matrix for baselines is set to have initial diagonal value matching $\sigma = 10$ for CBO, while we use an exponential decay of the σ for CBO ensures the convergence behavior due to Remark 11.

The benchmark results are summarized in Figure 9. CBO outperforms CMA and MPPI by a large margin. In Figure 10, we visualize the trajectories generated by CBO, MPPI, and CMA, respectively. It can be observed that CBO is able to successfully navigate through the tunnel, avoiding all obstacles (note that we consider the agent as a point mass), and reaches the tunnel, whereas MPPI and CMA get stuck in their local minima and fail to reach the tunnel.

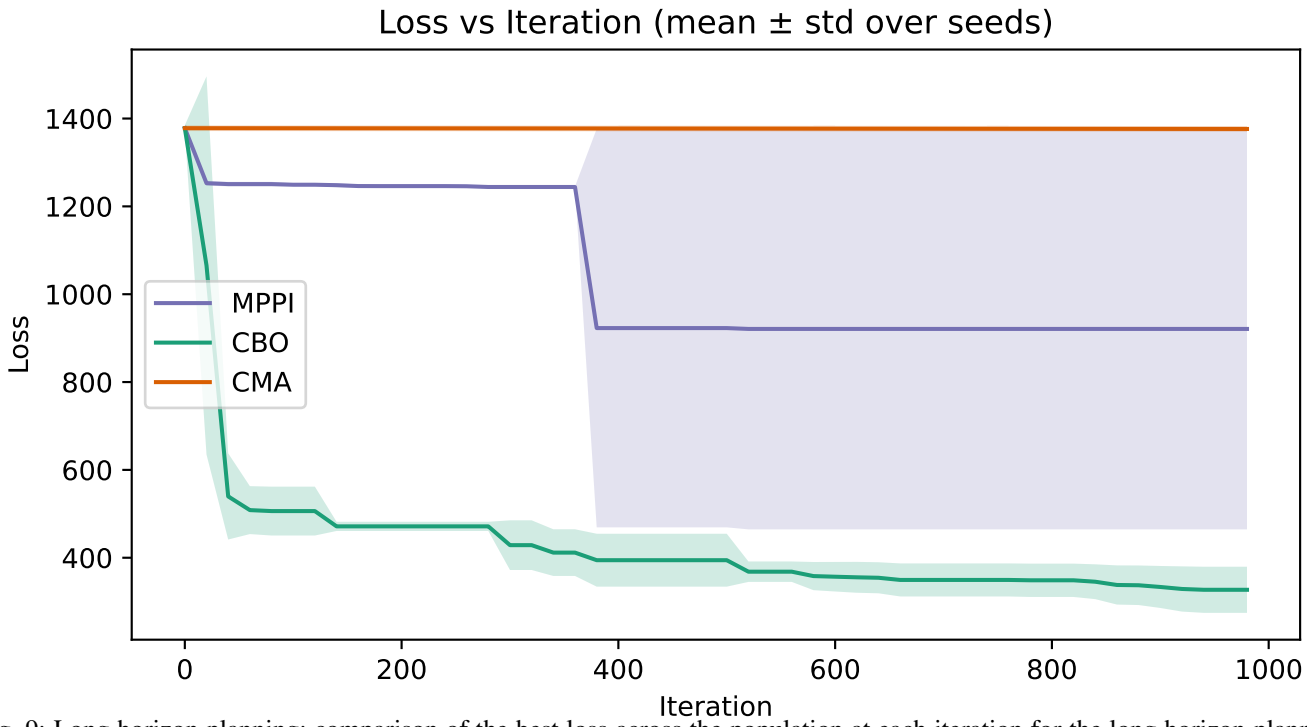


Fig. 9: Long horizon planning: comparison of the best loss across the population at each iteration for the long horizon planning problem. CBO wins by a large margin. CMA implemented according to (6) and (7).

B. Details on double cartpole experiment

The double cartpole (see an illustration in Figure 11) has the cart that is allowed to move within the range $[-3.8m, +3.8m]$ and the pole with the cart forming a revolute joint and another pole connected to the base pole via a revolute joint. The generalized coordinates are $\begin{bmatrix} q & \theta_1 & \theta_2 \end{bmatrix}^T$, where q is the cart position along the rail, θ_i with $i = 1, 2$ are the pole angles.

The initial state is

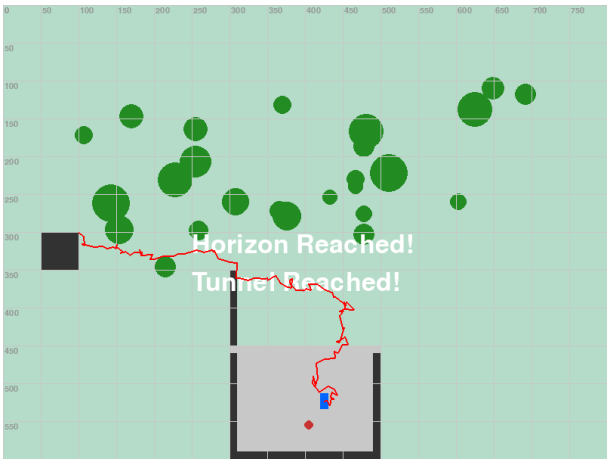
$$x_0 = \begin{bmatrix} q_0 \\ \dot{q}_0 \end{bmatrix} = \begin{bmatrix} q & \theta_1 & \theta_2 & \dot{q} & \dot{\theta}_1 & \dot{\theta}_2 \end{bmatrix}^T_{t=0} = 0, \tag{67}$$

where $\theta_i = 0$ corresponds to the pole hanging downward, and the upright configuration is $\theta_i = \pi$. Slider joint (cart) damping is set to $d_x = 10^{-4}$. To make the problem more challenging, we set friction loss to 0 for both revolute joints. The running cost is set to be the bound violation cost of control according to [17] and the terminal cost is calculated as the summation of all the following terms with equal weighting:

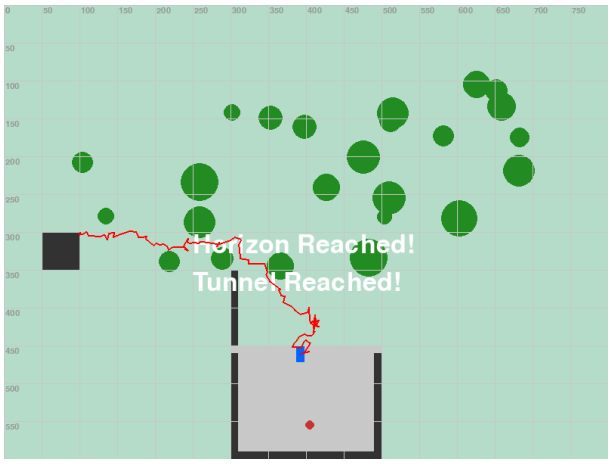
- distance to upright pole angles, i.e., for $i = 1, 2$, minimize $(\cos(\theta_i + \pi) - 1)^2 + (\sin(\theta_i + \pi))^2$.
- the distance of the cart to the center of the rail.
- the angular velocity of each pole.
- control effort over the entire horizon.

The control input is the force applied to the cart with a range of $20N[-u_{\max}, u_{\max}]$. To make the problem challenging, we deliberately set $u_{\max}$ as small as 0.5 to increase the control difficulty, while we set a long horizon of $T = 100$ zero-order hold control signal for 16 seconds. Population size $N = 5000$.

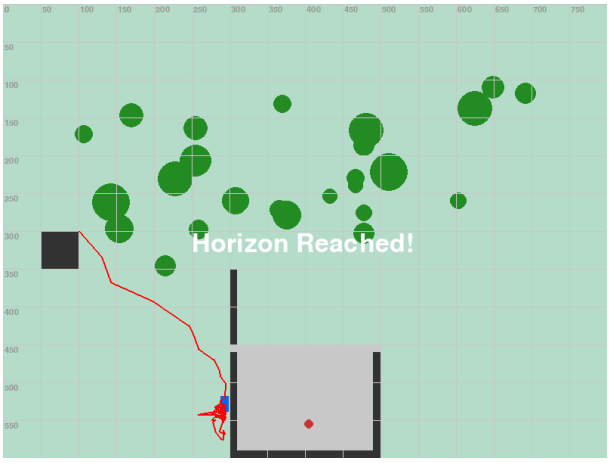
In Figure 12, we present benchmark results of loss minimization among CBO, MPPI and CMA-ES (code from [18]) where CBO performs the best. CMA-ES experiences large fluctuations of cost during the iteration, while CBO consistently generates good performance.



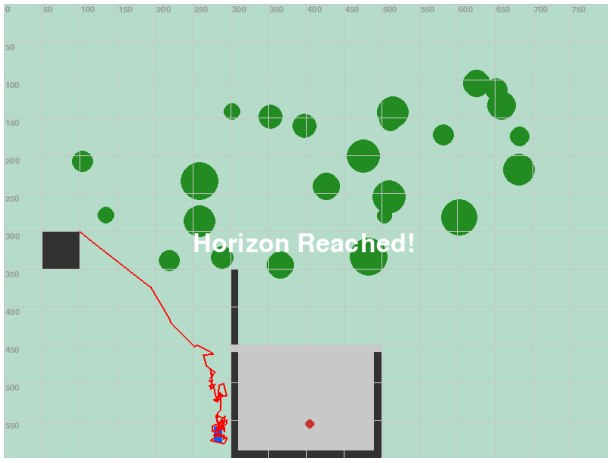
(a) CBO



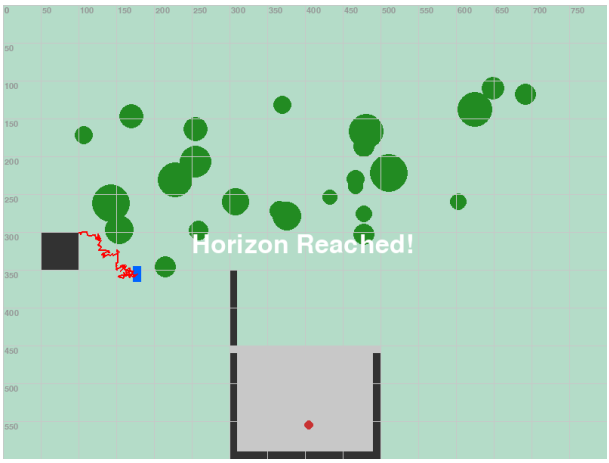
(b) CBO



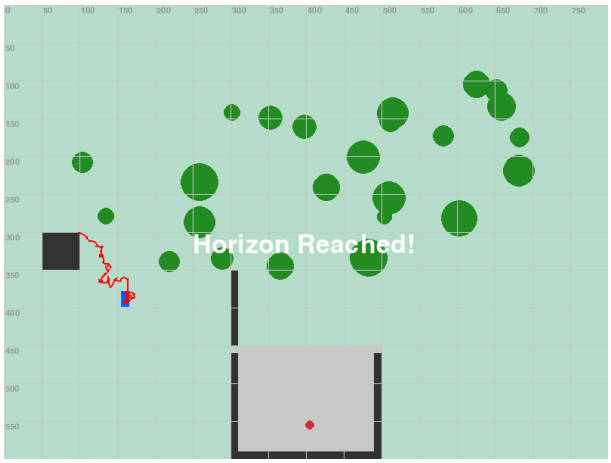
(c) MPPI



(d) MPPI



(e) CMA



(f) CMA

Fig. 10: MPPI and CMA get stuck at a local minimum and fail to reach the tunnel. CBO succeeds in circumventing the left wall barriers and all obstacles (point mass agent) and reaches the tunnel. Each image shows the trajectory of a single experiment for different methods. Each column corresponds to one environment setting (spatial distribution of obstacles).

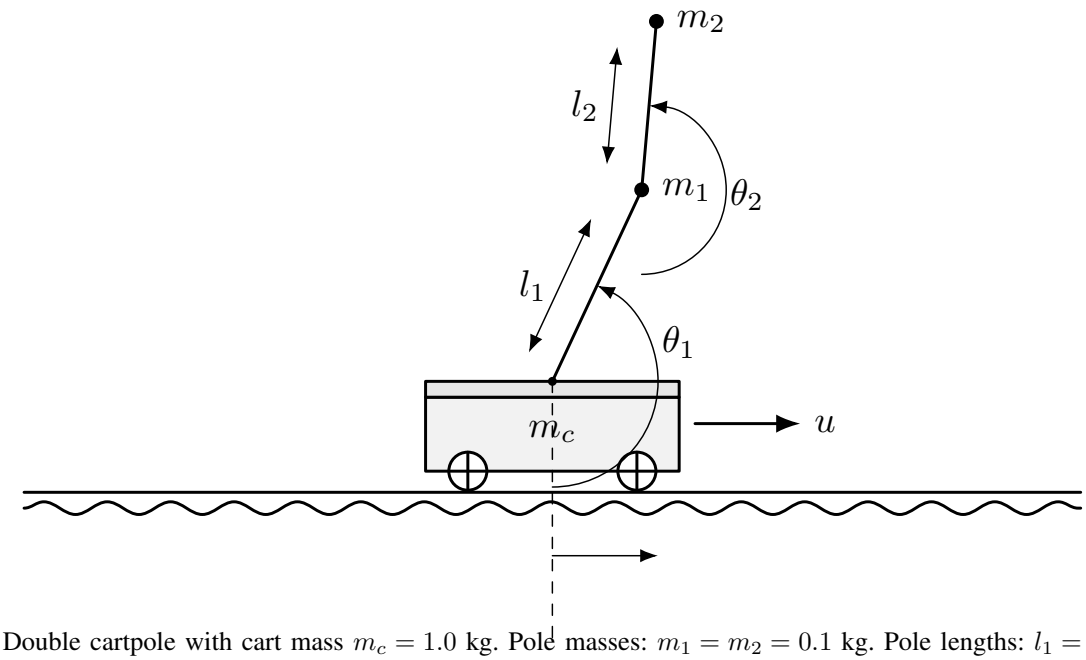


Fig. 11: Double cartpole with cart mass $m_c = 1.0$ kg. Pole masses: $m_1 = m_2 = 0.1$ kg. Pole lengths: $l_1 = l_2 = 1.0$m.

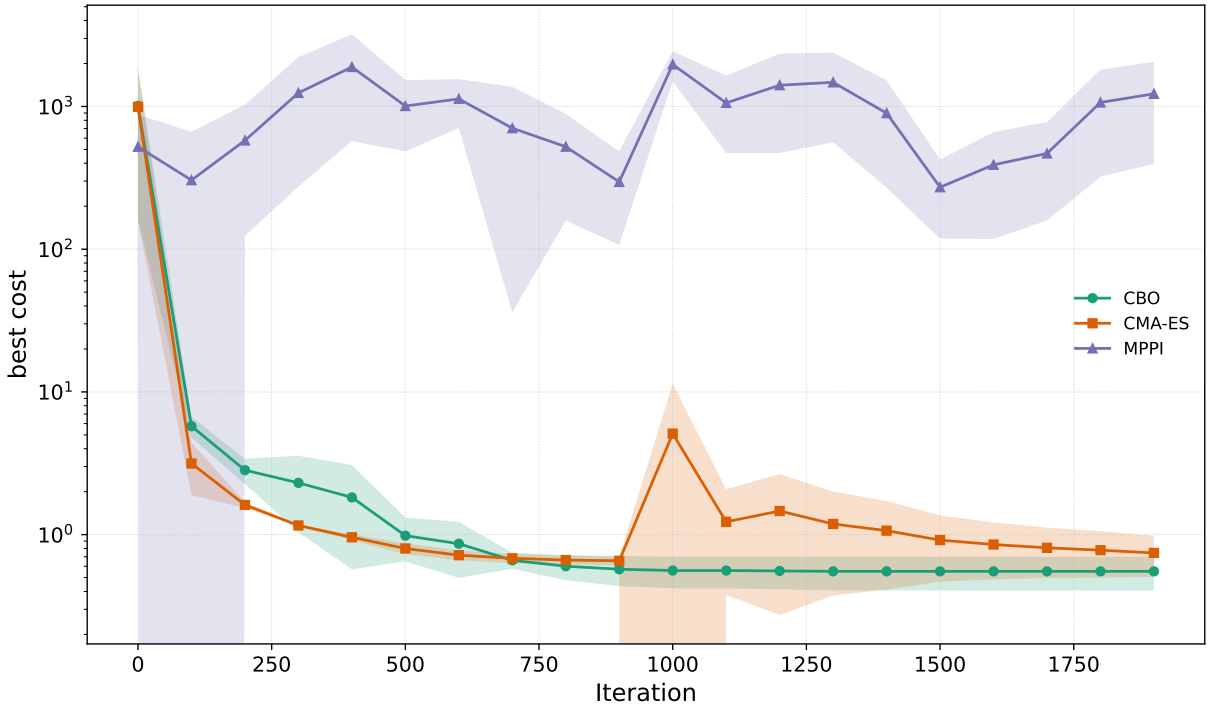


Fig. 12: Benchmark double-cartpole: On average, CBO performs the best with relatively small variance. CMA-ES is from implementation [18].