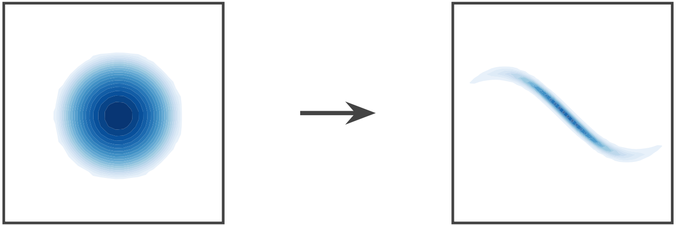
Learning and
Intelligent
Systems

TECHNISCHE
UNIVERSITÄT
BERLIN

sakana.ai

Stein Variational Evolution Strategies

Cornelius V. Braun, Robert T. Lange, Marc Toussaint



Conference on Uncertainty in Artificial Intelligence, 24th July 2025

Data Generation

1 Motivation

Data drives modern ML applications:

Robotics

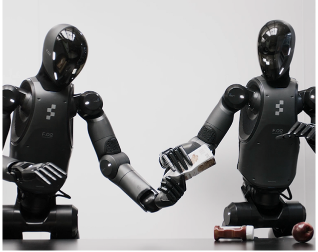
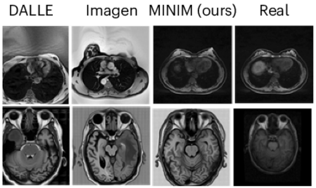


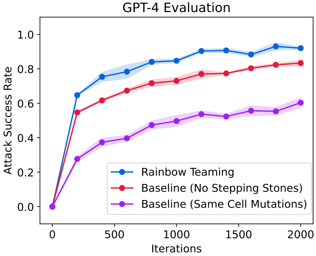
Figure AI 01 Robot.

Medical ML



Wang et al., “Self-improving generative foundation model for synthetic medical image generation and clinical applications”, *Nature* (2025).

Model Alignment



Samvelyan et al., “Rainbow Teaming: Open-Ended Generation of Diverse Adversarial Prompts”, *NeurIPS* (2024).

Data Generation

- Generative models require **a lot of training data.**
- Synthetic data generation via **optimization** is important for scientific discovery and model performance.
- **Goal:** find data points that are diverse and perform well.

→ **Inference** as a principled way to generate **diverse** solutions to optimization problems.

Optimization and Inference

Setting: Given objective function $f : \mathbb{R}^d \to \mathbb{R}$, that we want to **minimize**, we construct the following posterior:

$$p(x) = \frac{e^{-f(x)}}{Z}, \quad \text{with } \textbf{unknown} \quad Z = \int_{\mathbb{R}^d} e^{-f(x)} \, \mathrm{d}x.$$

Goal: Construct distribution q that matches p as well as possible, i.e.

$$\min_{q \in \mathcal{Q}} D_{\mathrm{KL}}(q \,\|\, p).$$

Stein Variational Gradient Descent

Idea: Construct a map T that gradually pushes samples towards p:

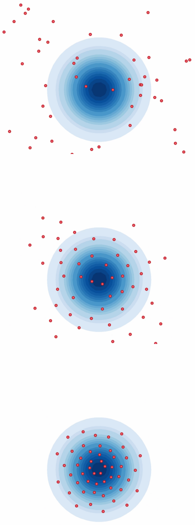
$$x_{t+1} = T(x_t) = x_t + \epsilon\phi(x_t)$$

where ϵ is step size and ϕ is a vector field.

Theorem: SVGD Update[1]

For RKHS with kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, the optimal step $\phi^*(\cdot)$ is given by

$$\phi^*(x) = \mathbb{E}_{y\sim q}[\underbrace{\nabla_y \log p(y)k(x,y)}_{\textit{driving force}} + \underbrace{\nabla_y k(x,y)}_{\textit{repulsive force}}]. \tag{1}$$



---

[1] Liu & Wang, "Stein variational gradient descent: A general purpose bayesian inference algorithm", *Neurips* (2016).

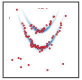
Challenges in Practice

$\cancel{\nabla f}$

1. No / bad gradients (e.g., robotics).
→ **Plain SVGD does not work.**

σ^2

2. MC gradient approximations have **high variance**.
→ **Slow convergence.**



3. Differentiable surrogates trade tractability for performance.
→ **Poor model fit.**

Challenges in Practice

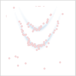
$\cancel{\nabla f}$

1. No / bad gradients (e.g., robotics).
$\rightarrow$ **Plain SVGD does not work.**

How to perform *gradient-free* inference efficiently?

σ^2

2. MC gradient approximations have **high variance**.
$\rightarrow$ **Slow convergence.**

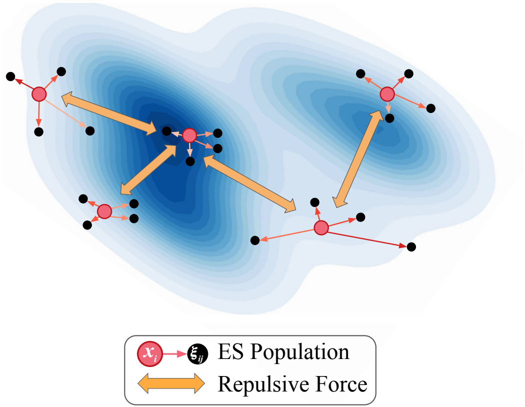


3. Differentiable surrogates trade tractability for performance.
$\rightarrow$ **Poor model fit.**

Method Overview

3 Stein Variational CMA-ES (SV-CMA-ES)

- Combine **evolution strategies (ES)** with **SVGD** into single update.
- Each **SVGD particle** represents **ES population**.
- **Estimate driving force** using ES.
- Use **analytic gradients** for repulsion from SVGD.
- ES updates add **momentum-like** mechanism based on step-size adaptation.



Evolution Strategies

3 Stein Variational CMA-ES (SV-CMA-ES)

CMA-ES[2] (informal)

- Update Gaussian search distribution $\mathcal{N}(x, \sigma^2 C)$ to maximize likelihood
- Mean x (max. likelihood estimate):

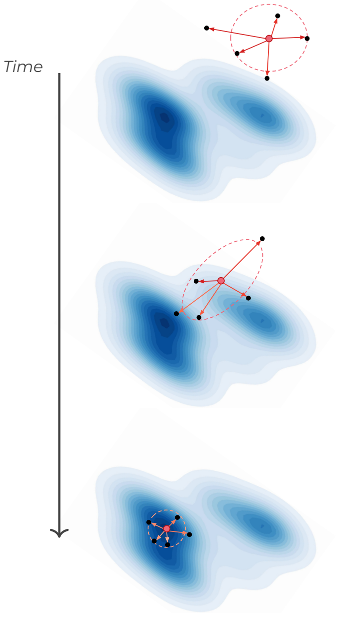
$$x \leftarrow x + \sum_{i=1}^{m} w_i(\xi_i - x),\ \xi_i \sim \mathcal{N}(x, \sigma^2 C).$$

- Covariance C:

$$C \leftarrow \bar{\alpha} C + \alpha_1 p_c {p_c}^T + \alpha_m \sum_{i=1}^{n} \bar{w}_i y_i y_i^T$$

- Step-size σ:

$$\sigma \leftarrow \sigma \times \exp\left(\frac{\alpha_\sigma}{d_\sigma}\left(\frac{\|p_\sigma\|}{\mathbb{E}\|\mathcal{N}(0,\mathbf{I})\|} - 1\right)\right)$$



---

[2] Hansen & Ostermeier. "Completely derandomized self-adaptation in evolution strategies", *Evolutionary computation* (2001).

SV-CMA-ES Particle Update

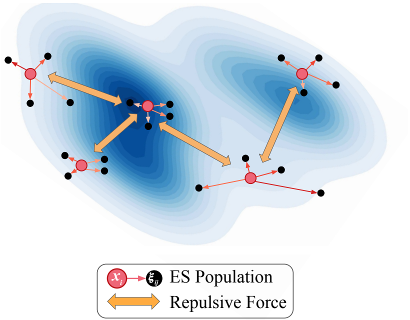
- Run separate search distribution for each SVGD particle and update $x_i \leftarrow x_i + \phi(x_i)$.

- Update each particle by combining the CMA-ES mean update and the kernel-based repulsion:

$$\phi(x_i) = \mathbb{E}_{x_j \sim q}\Big[k(x_j, x_i)\Delta x_{j\,\text{CMA}} + \nabla_{x_j} k(x_j, x_i)\Big]$$

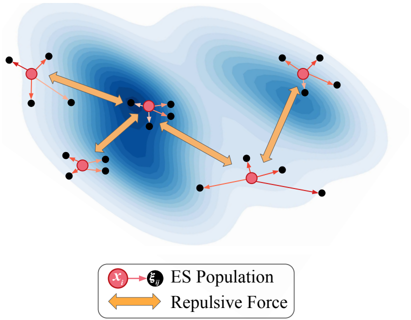
$$\approx \frac{1}{\varrho}\sum_{j=1}^{\varrho}\left[\underbrace{\left[\sum_{\ell=1}^{m} w_{j\ell}(\xi_{j\ell} - x_j)\right] k(x_j, x_i)}_{\textit{driving force}} + \underbrace{\nabla_{x_j} k(x_j, x_i)}_{\textit{repulsive force}}\right]$$

- Update remaining search distribution parameters following CMA-ES.



SV-CMA-ES Particle Update

Problem 1: Weighted average in driving force reduces particle-steps, and CMA-ES step-size will shrink.

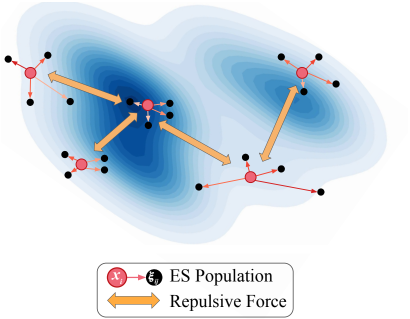


Problem 2: Limited exploration.

SV-CMA-ES Particle Update

Problem 1: Weighted average in driving force reduces particle-steps, and CMA-ES step-size will shrink.

→ **Replace smoothing in driving force by local step estimate at particle.**[3]



Problem 2: Limited exploration.

---

[3] D'Angelo, Fortuin, & Wenzel, "On stein variational neural network ensembles", *arXiv* (2021).

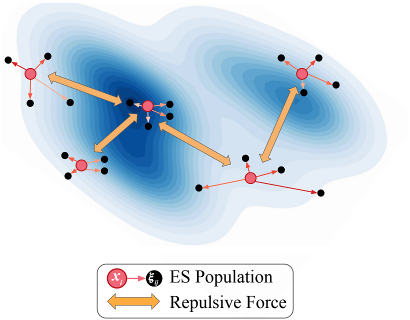
SV-CMA-ES Particle Update

Problem 1: Weighted average in driving force reduces particle-steps, and CMA-ES step-size will shrink.

→ **Replace smoothing in driving force by local step estimate at particle.**[3]

Problem 2: Limited exploration.

→ **Add annealing term** $\gamma : [0, 1] \to \mathbb{R}$**.**



---

[3] D'Angelo, Fortuin, & Wenzel, "On stein variational neural network ensembles", *arXiv* (2021).

SV-CMA-ES

SV-CMA-ES Update

1. Sample ϱ populations of size m from the respective search distributions: $\{\xi_{i1}, \dots, \xi_{im}\} \sim \mathcal{N}(x_i, \sigma_i^2 C_i)$.

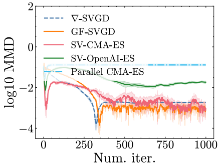
2. Update population means x_i:

$$\phi(x_i) = \underbrace{\sum_{\ell=1}^{m} w_{i\ell}(\xi_{i\ell} - x_i)}_{\textit{driving force}} + \underbrace{\frac{\gamma(t)}{\varrho} \sum_{j=1}^{\varrho} \nabla_{x_j} k(x_j, x_i)}_{\textit{repulsive force}} . \tag{2}$$

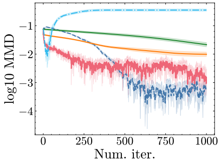
3. Given $\Delta x_i = x_i + \phi(x_i)$, update step size σ_i and C_i based on CMA-ES.

Synthetic Density Sampling

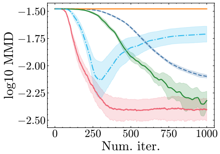
4 Experiments



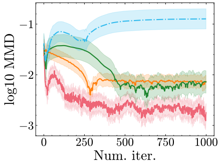
(a) Gaussian Mixture[4]



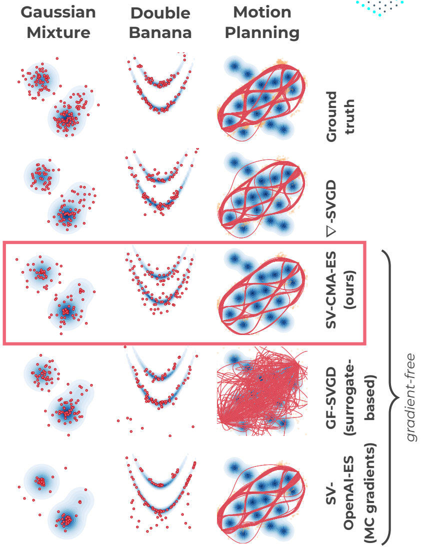
(b) Double Banana



(c) Motion Planning



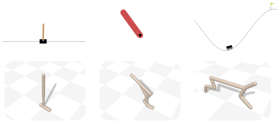
(d) MMD w.r.t. ∇-SVGD



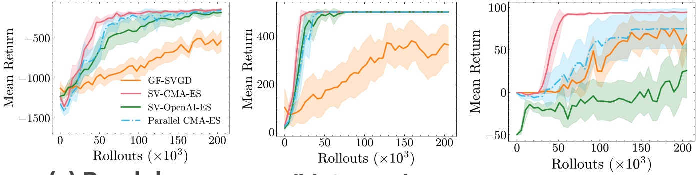
---

[4] MMD = Maximum Mean Discrepancy w.r.t. Ground Truth

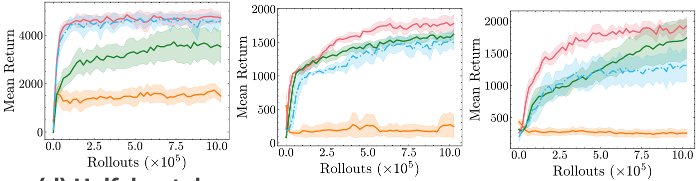
RL Controller Optimization



- Classic Control Tasks
- Direct Policy Search
- Domain = MLP parameters → High dimensional



(a) Pendulum (b) Cartpole (c) Mountaincar



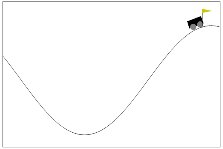
(d) Halfcheetah (e) Hopper (f) Walker

Mountaincar Controller Optimization

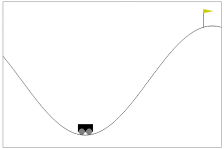
- Penalties for controls
- Sparse reward for reaching goal

→ Task is difficult due to local optimum.

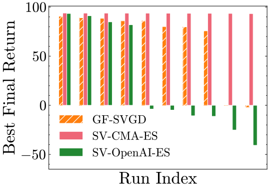
Diversity pays off!



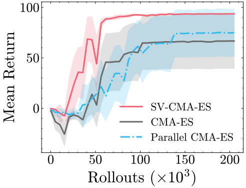
SV-CMA-ES solves task consistently.



Baselines fail to solve task consistently.



Per-seed results.



Comparison to CMA-ES-based alternatives.

Summary

5 Summary

- **SV-CMA-ES: Combine CMA-ES with SVGD** for gradient-free sampling & optimization
- **Faster** convergence than prior methods
- **More diverse** solutions than prior methods

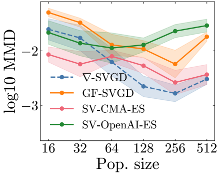
Come chat with me: **Poster today at 4pm**

arXiv:2410.10390

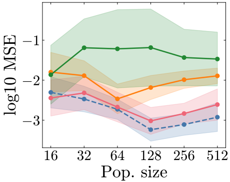
amazon robotics

We kindly thank the Amazon Fulfillment Technologies and Robotics team for their financial support.

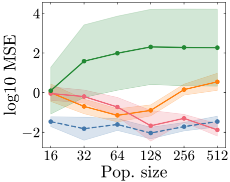
Scaling Experiments



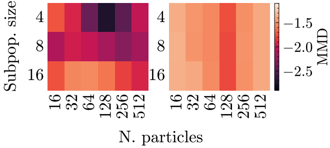
(a) MMD w.r.t. ground truth



(b) Estimate $\mathbb{E}[x]$

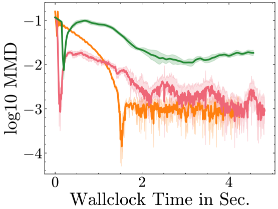


(c) Estimate $\mathbb{V}[x]$

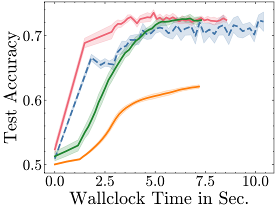


(d) Subpop. Size Scaling

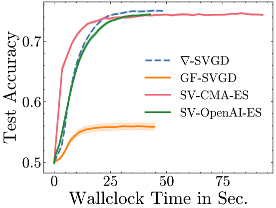
Runtime Comparison



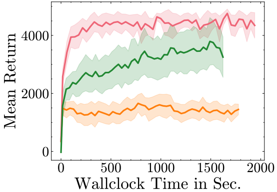
(a) Gauss. Mix. Sampling



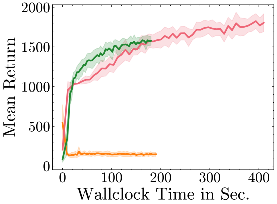
(b) Credit Log. Reg.



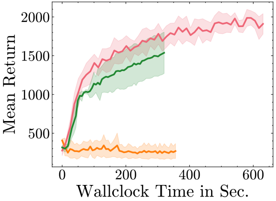
(c) Covtype Log. Reg.



(d) Halfcheetah RL



(e) Hopper RL



(f) Walker RL