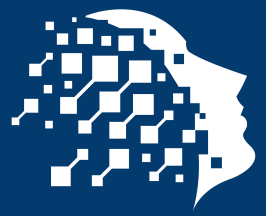
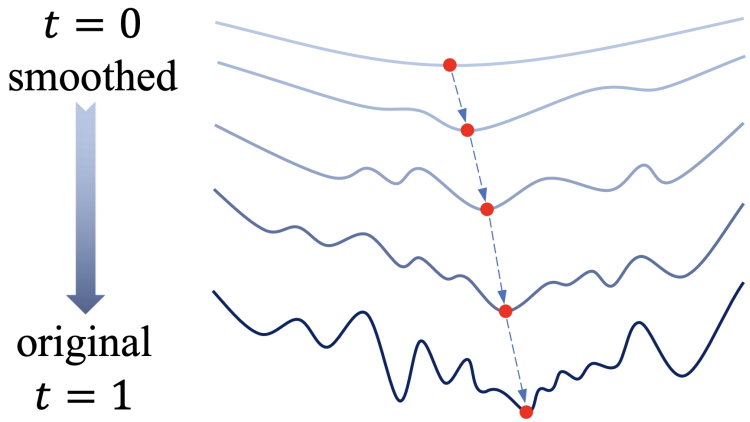



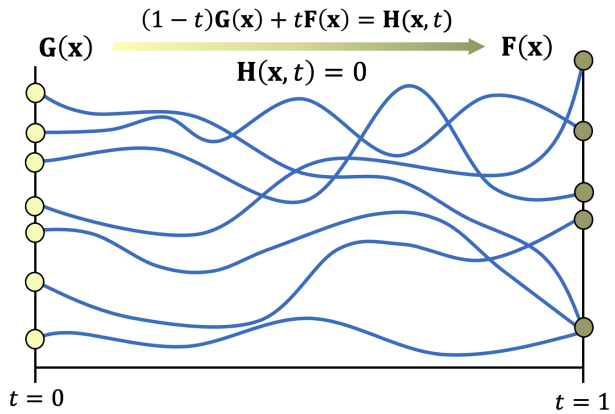
Neural Predictor-Corrector: Solving Homotopy Problems with Reinforcement Learning

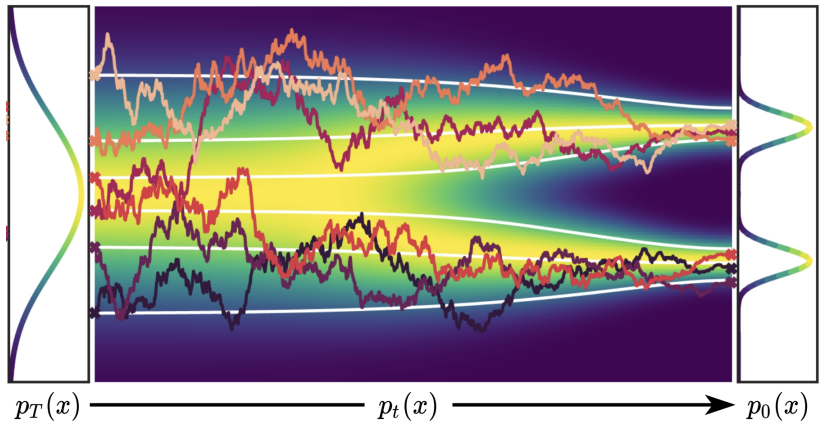
Jiayao Mai[1*], Bangyan Liao[2*], Zhenjun Zhao[3], Yingping Zeng[1],
Haoang Li[4], Javier Civera[3], Tailin Wu[2], Yi Zhou[1†], Peidong Liu[2†]

https://github.com/maijiayao1/NPC

Background: Homotopy Paradigm





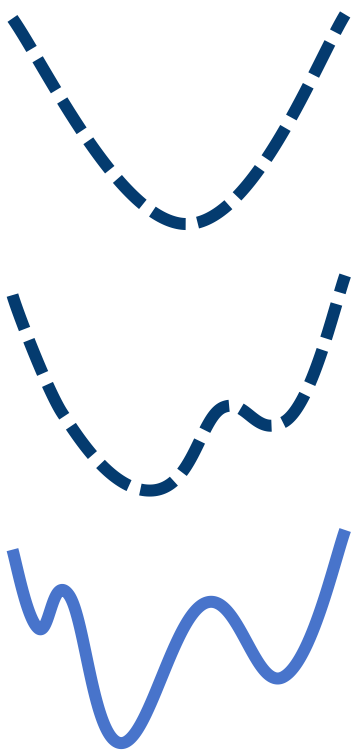


Optimization[1] **Polynimial Root Finding**[2] **Sampling**[3]

A general principle for solving challenging problems

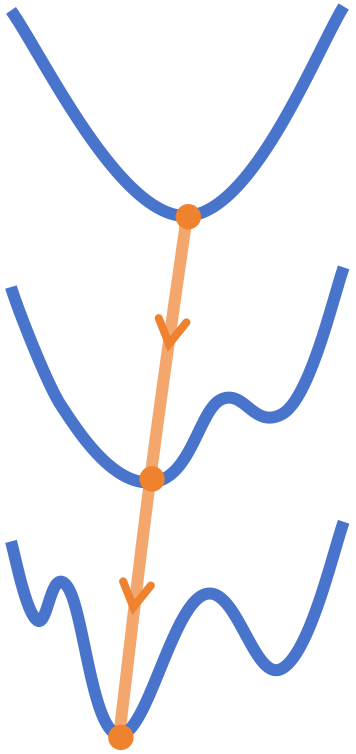
[1] Lin X, Yang Z, Zhang X, et al. Continuation path learning for homotopy optimization[C]//International Conference on Machine Learning. PMLR, 2023: 21288-21311.
[2] Zhang, Xinyue, et al. "Simulator HC: Regression-based Online Simulation of Starting Problem-Solution Pairs for Homotopy Continuation in Geometric Vision." *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2025.
[3] Song Y, Sohl-Dickstein J, Kingma D P, et al. Score-based generative modeling through stochastic differential equations[J]. arXiv preprint arXiv:2011.13456, 2020.

Background: Homotopy Paradigm



Easy Problem
Known Solutions

Hard Problem
Unknown Solutions



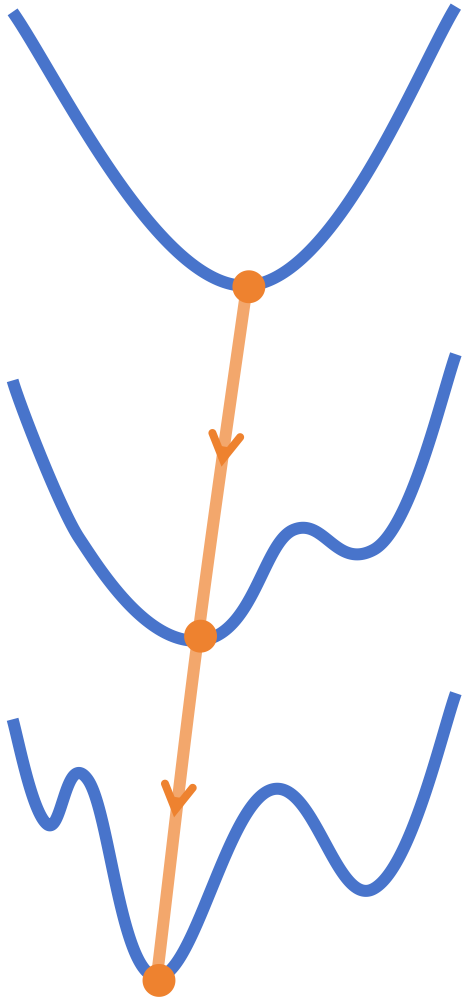
Explicit Homotopy
Interpolation Construction

Implicit Solution
Trajectory Tracing

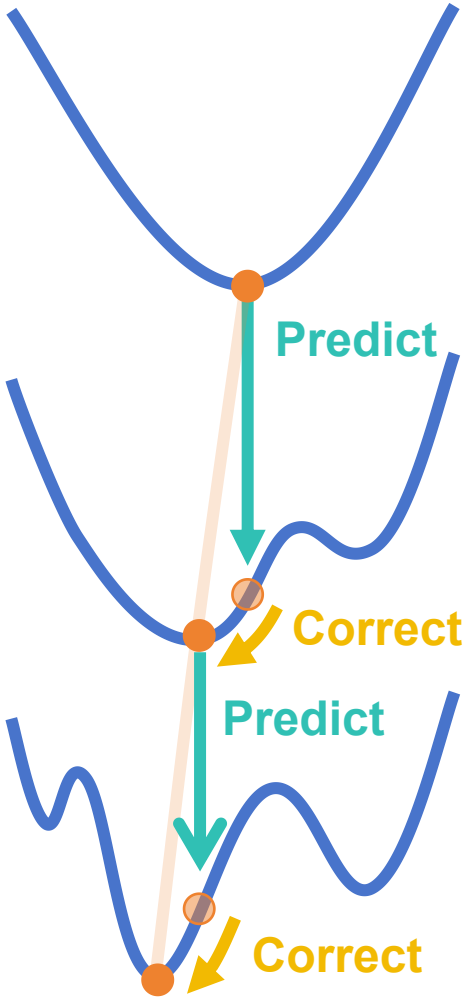
Our Scope

Motivation: Classical Predictor-Corrector

Implicit Solution Trajectory



Classical Predictor-Corrector

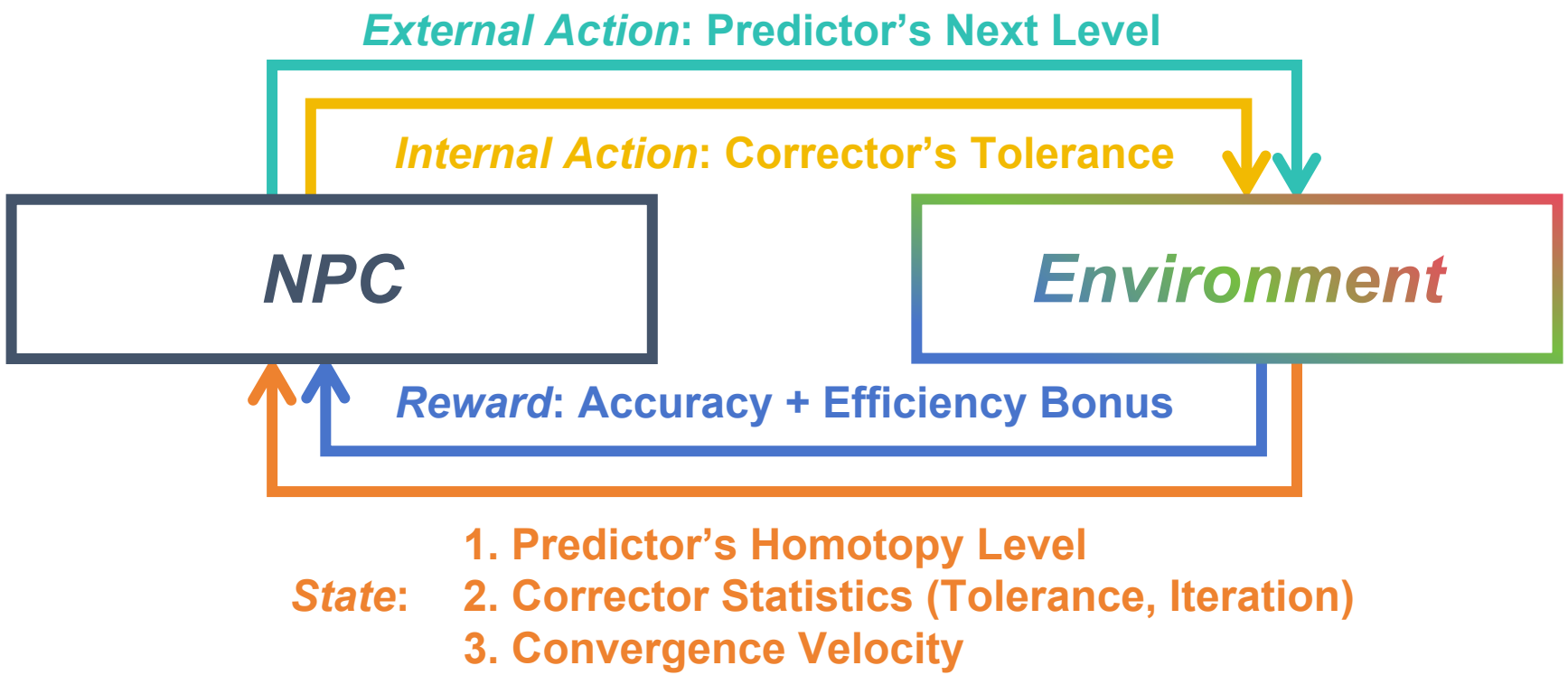


Algorithm

while $t_n < T$ **do**
  Predictor: Predict next level t_n
  Predictor: Predict $\mathbf{x}_{t_n}$
  While $H(\mathbf{x}_{t_n}, t_n) \leq \epsilon$ **do**
    Corrector: Correct $\mathbf{x}_{t_n}$
  end
end

Classical PC relies on hand-crafted heuristics

Our Method: Neural Predictor-Corrector (NPC)



NPC can find the optimal policy

Our Method: Neural Predictor-Corrector (NPC)

ICLR

Classical Predictor-Corrector

Algorithm 1

while $t_n < T$ **do**
  Predictor: Predict next level t_n
  Predictor: Predict $\mathbf{x}_{t_n}$
  While $H(\mathbf{x}_{t_n}, t_n) \leq \epsilon$ **do**
    Corrector: Correct $\mathbf{x}_{t_n}$
  end
end

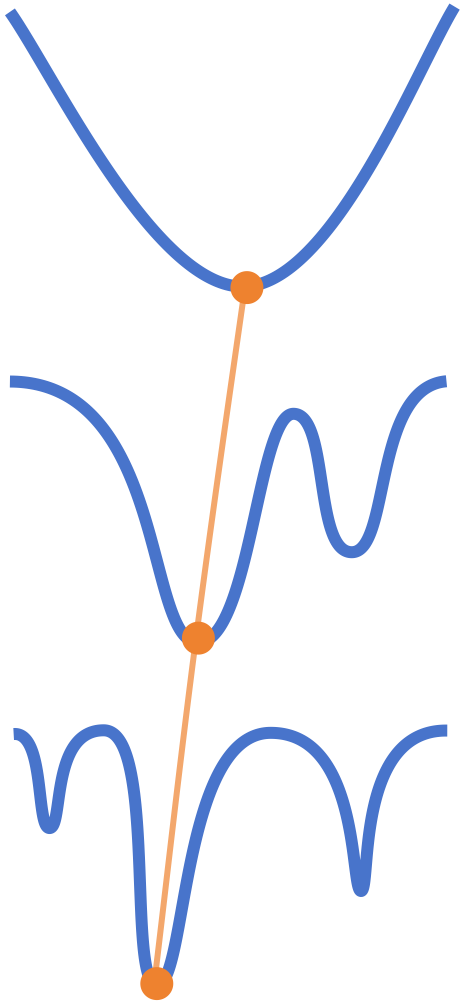
Neural Predictor-Corrector

Algorithm 2

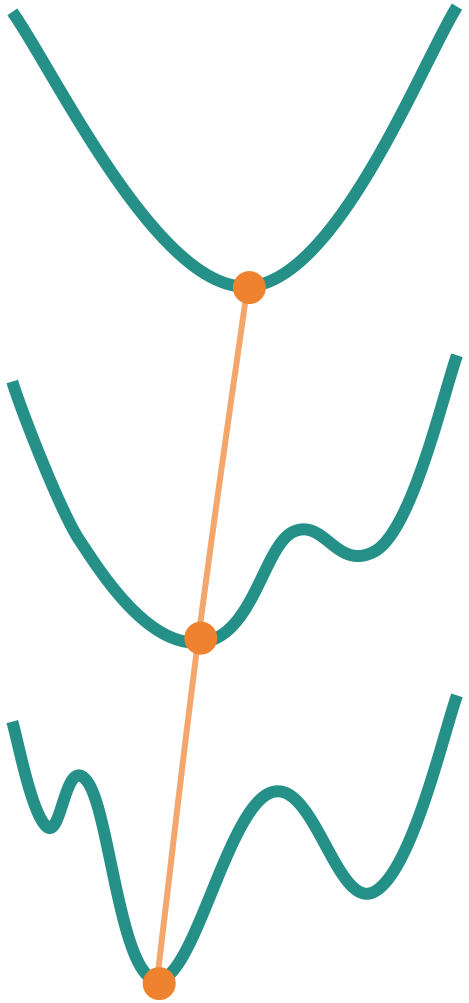
while $t_n < T$ **do**
  NPC: $\{\Delta t_n, \epsilon_n\} = \mathbf{NN}(t_{n-1}, \epsilon_{n-1}, \tau_{n-1})$
  Predictor: Predict next level $t_n = t_{n-1} + \Delta t_n$
  Predictor: Predict $\mathbf{x}_{t_n}$
  While $H(\mathbf{x}_{t_n}, t_n) \leq \epsilon_n$ **do**
    Corrector: Correct $\mathbf{x}_{t_n}$
  end
  Collect corrector statistics ϵ_n
  Collect convergence velocity τ_n
end

NPC can find the optimal policy

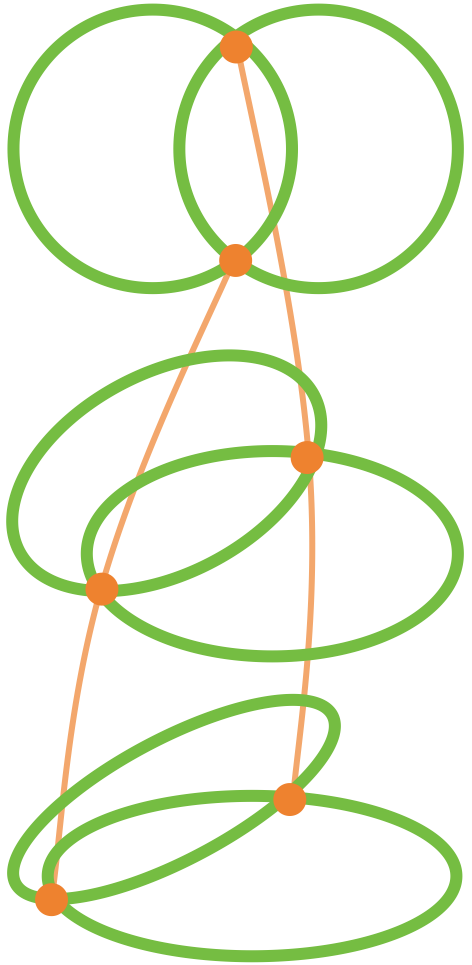
Experiments



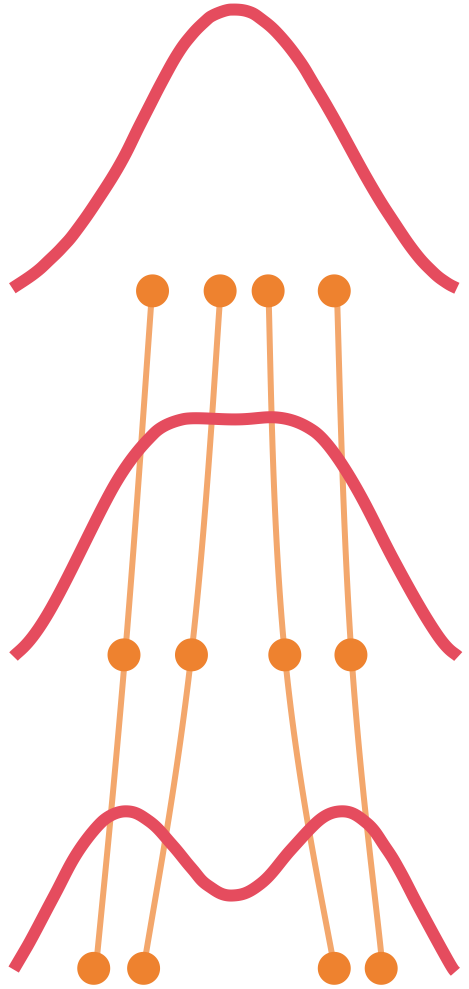
Robust Optimization



Global Optimization

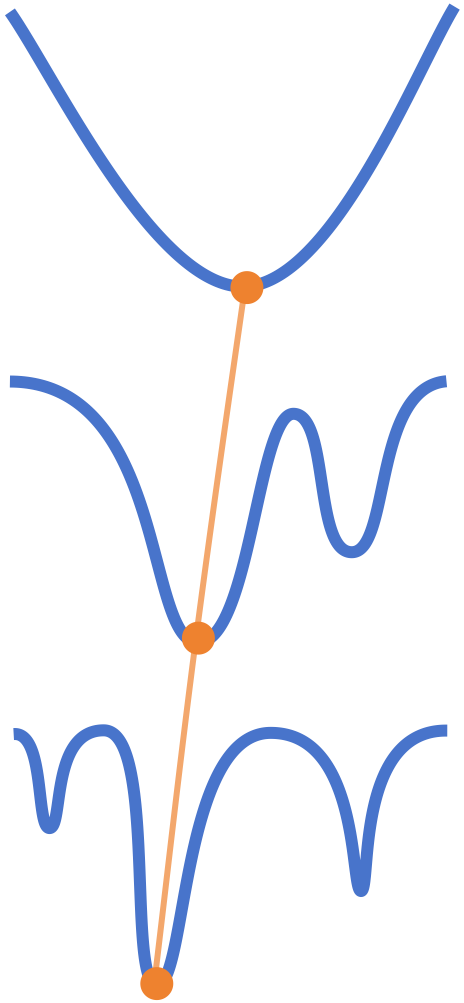


Polynimial Root Finding



Sampling

Problem 1 : Robust Optimization via GNC

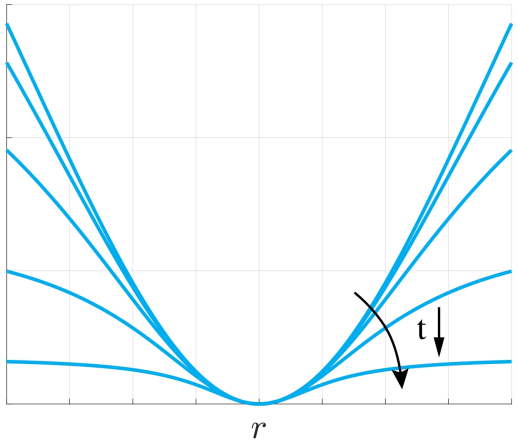


Robust Optimization

Graduated Non-Convexity (GNC)[1]

Outer Interpolation (Geman-McClure)

$$H(\mathbf{x}, t) = \sum_i \frac{\bar{c}^2 r(\mathbf{x}, y_i)^2}{\bar{c}^2 + t r(\mathbf{x}, y_i)^2}$$



Predictor

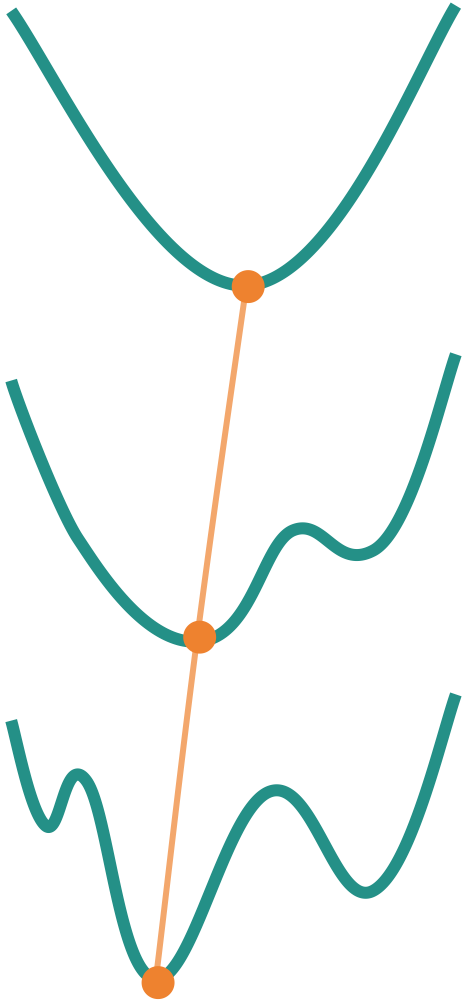
$$t_{i+1} = t_i + \Delta t$$
$$\mathbf{x}_{t_{i+1}} = \mathbf{x}^*_{t_i}$$

Corrector (*Gauss-Newton*)

$$\mathbf{x}^*_t = \arg\min_{\mathbf{x}} H(\mathbf{x}, t)$$

[1] Yang H, Antonante P, Tzoumas V, et al. Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection[J]. IEEE Robotics and Automation Letters, 2020, 5(2): 1127-1134.

Problem 2 : Global Optimization via GH

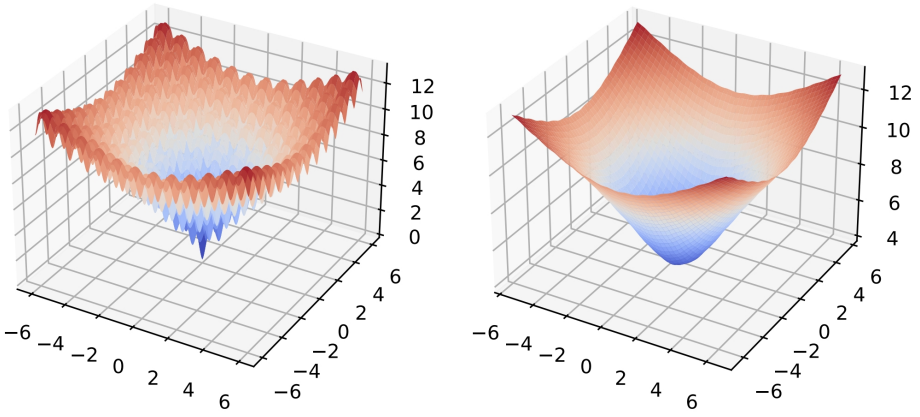


Global Optimization

Gaussian Homotopy (GH)

Outer Interpolation

$$H(\mathbf{x}, t) = g(\mathbf{x}) * \mathcal{N}(0, t\sigma^2)$$



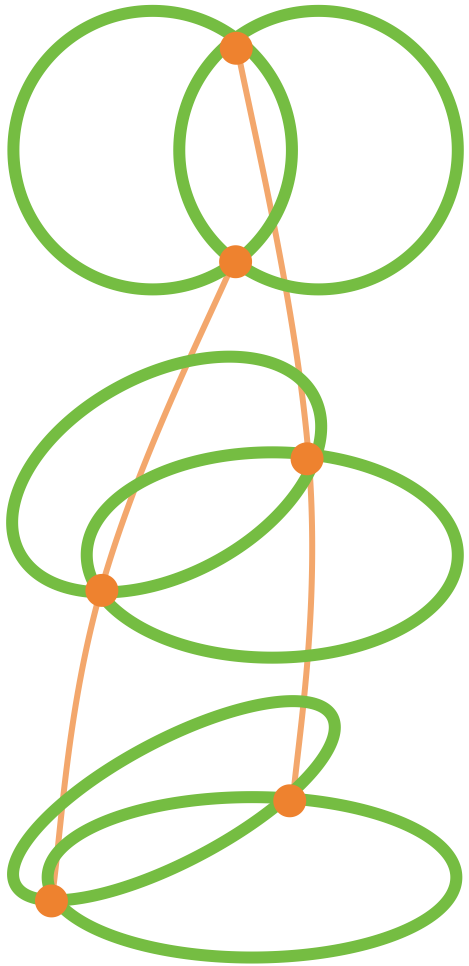
Predictor

$$t_{i+1} = t_i + \Delta t$$
$$\mathbf{x}_{t_{i+1}} = \mathbf{x}^*_{t_i}$$

Corrector

$$\mathbf{x}^*_t = \arg\min_{\mathbf{x}} H(\mathbf{x}, t)$$

Problem 3 : Polynomial Root-Finding via HC

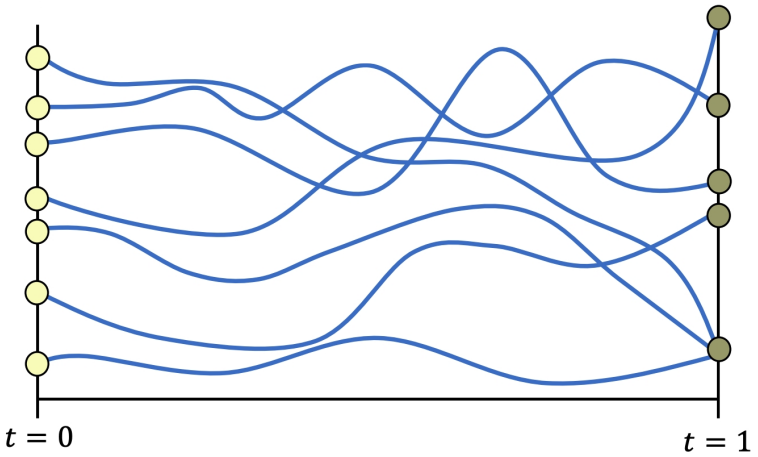


Polynimial Root Finding

Homotopy Continuation (HC)[1]

Outer Interpolation

$$H(\mathbf{x}, t) = g(\mathbf{x})(1-t) + f(\mathbf{x})t$$



Predictor

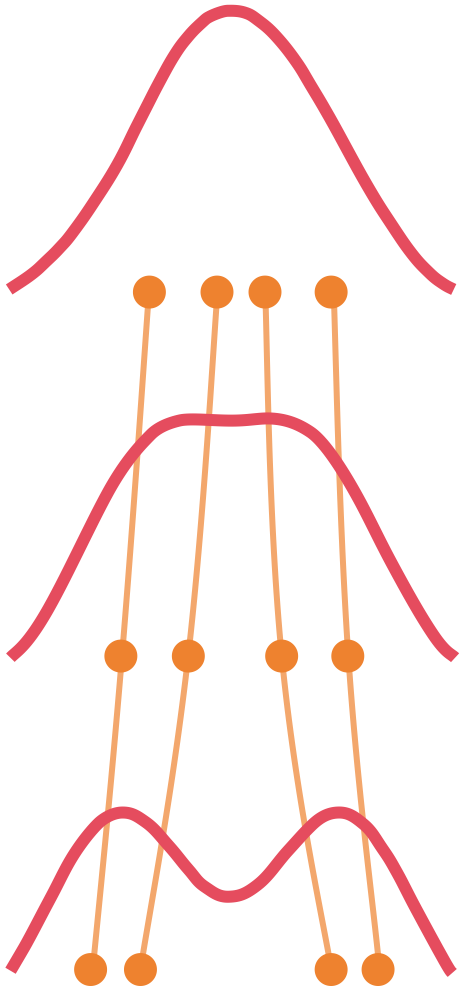
$$t_{i+1} = t_i + \Delta t$$
$$\{\mathbf{x}_{t_{i+1}}\} = \{\mathbf{x}^*_{t_i} + \partial_t H(\mathbf{x}^*_{t_i}, t_i)\Delta t\}$$

Corrector (*Gauss-Newton*)

$$\{\mathbf{x}^*_t\} = \{H(\mathbf{x}, t) = 0\}$$

[1] Bates D J, Sommese A J, Hauenstein J D, et al. Numerically solving polynomial systems with Bertini[M]. Society for Industrial and Applied Mathematics, 2013.

Problem 4 : Sampling via ALD

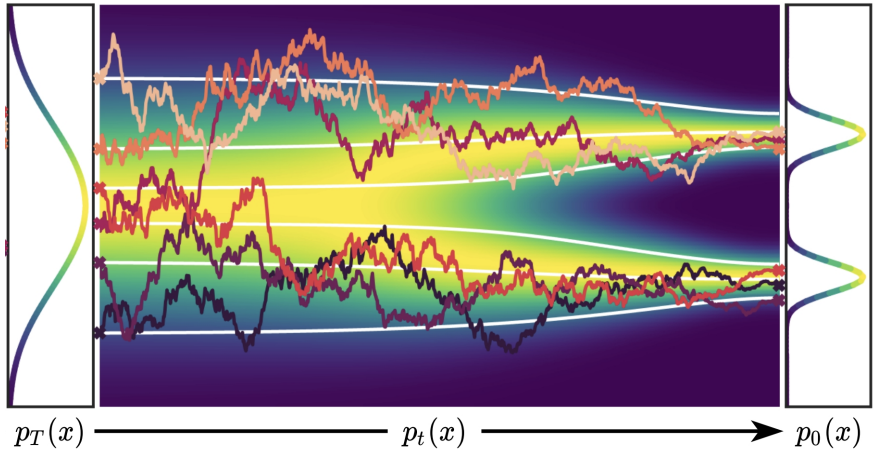


Sampling

Annealed Langevin Dynamics (ALD)[1]

Outer Interpolation

$$H(\mathbf{x}, t) \propto \exp(-g(\mathbf{x})(1-t) - f(\mathbf{x})t)$$



Predictor

$$t_{i+1} = t_i + \Delta t$$
$$\{X^0_{t_{i+1}}\} = \{X^*_{t_i}\}$$

Corrector (*LD*)

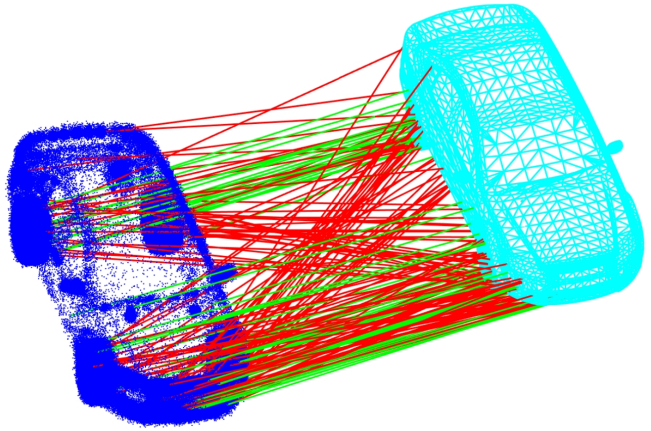
$$\mathrm{d}X^s_t = \nabla log\ H(\mathbf{x}, t)\ \mathrm{d}s + \mathrm{d}W_s$$

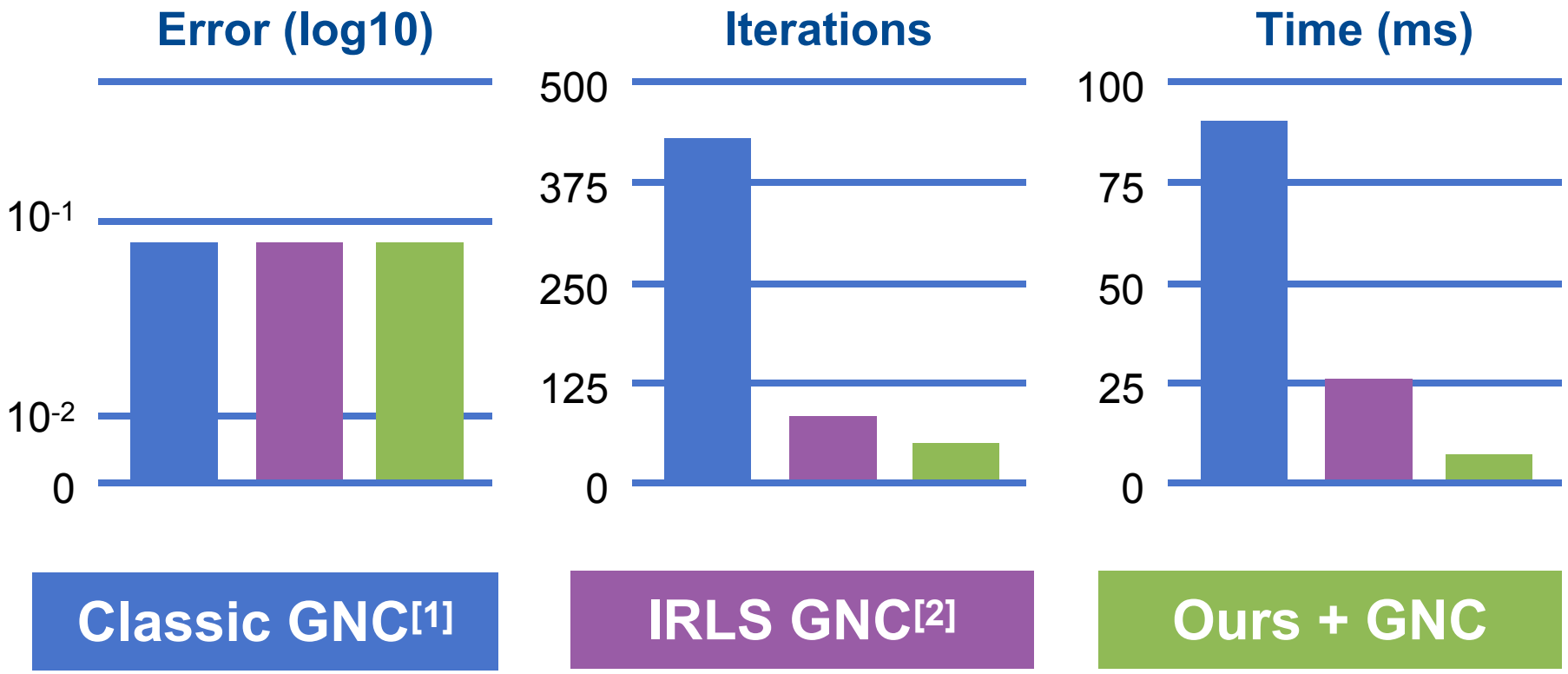
[1] Song Y, Sohl-Dickstein J, Kingma D P, et al. Score-based generative modeling through stochastic differential equations[J]. arXiv preprint arXiv:2011.13456, 2020.

Problem 1 : Robust Optimization via GNC

ICLR

Point Cloud Registration





Same Performance, Less Iterations, Less Time

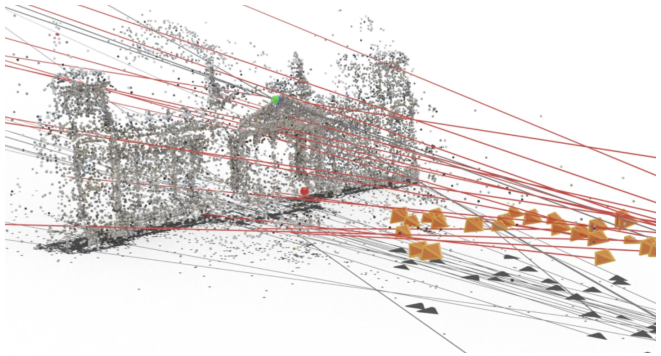
[1] Yang H, Antonante P, Tzoumas V, et al. Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection[J]. IEEE Robotics and Automation Letters, 2020, 5(2): 1127-1134.

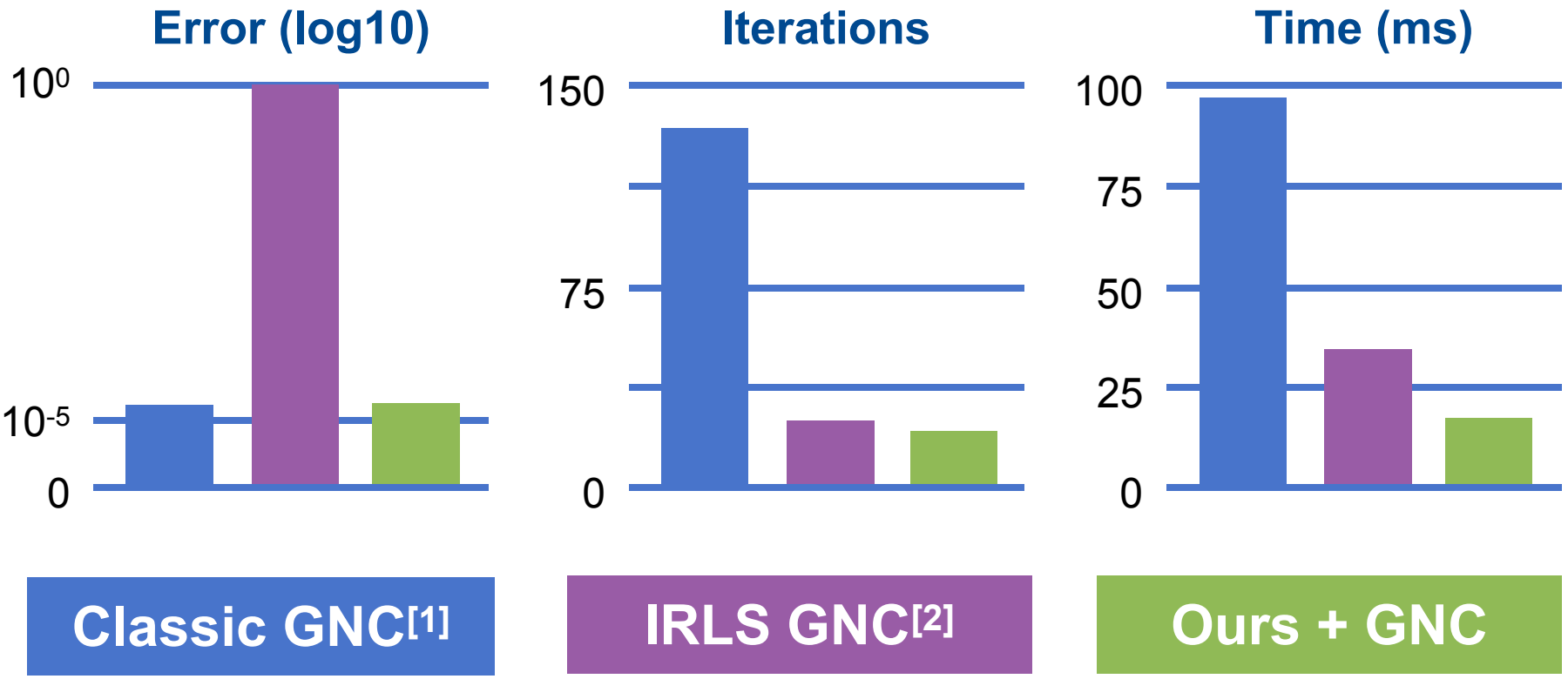
[2] Peng L, Kümmerle C, Vidal R. On the convergence of IRLS and its variants in outlier-robust estimation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 17808-17818.

Problem 1 : Robust Optimization via GNC

ICLR

Multi-view Triangulation



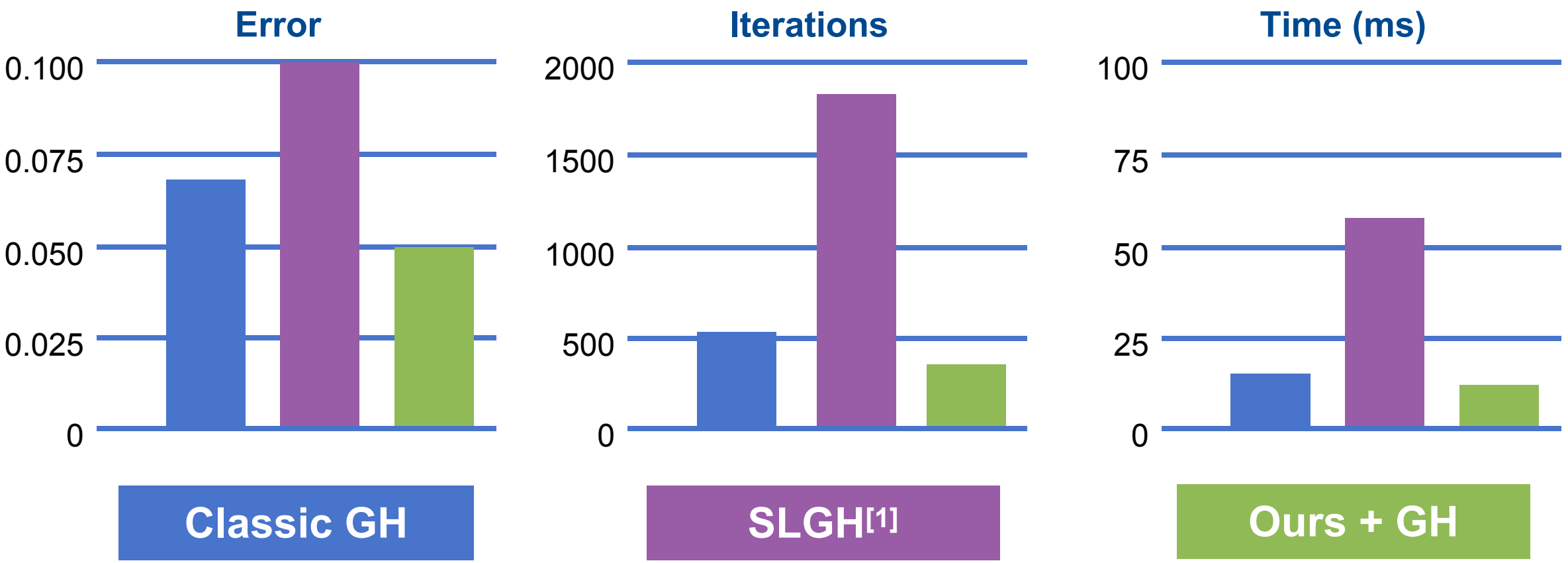


Less Error, Less Iterations, Less Time

[1] Yang H, Antonante P, Tzoumas V, et al. Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection[J]. IEEE Robotics and Automation Letters, 2020, 5(2): 1127-1134.

[2] Peng L, Kümmerle C, Vidal R. On the convergence of IRLS and its variants in outlier-robust estimation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 17808-17818.

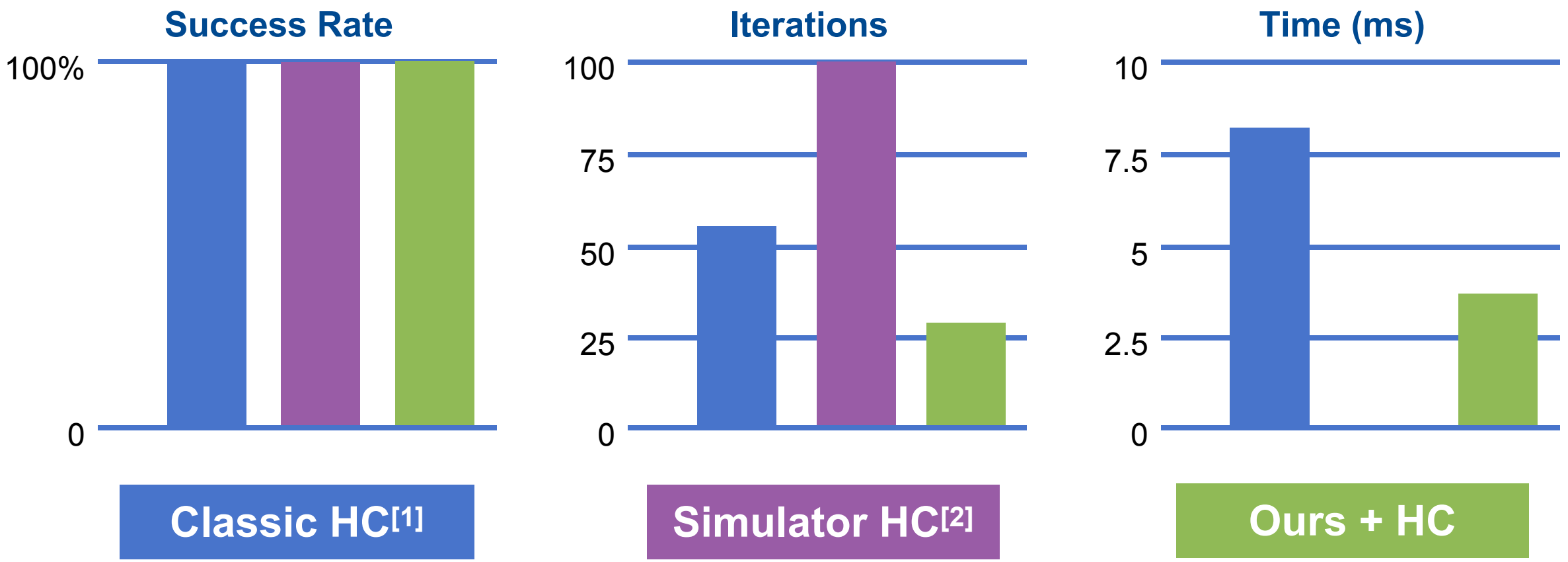
Problem 2 : Global Optimization via GH



Less Error, Less Iterations, Less Time

[1] Iwakiri H, Wang Y, Ito S, et al. Single loop gaussian homotopy method for non-convex optimization[J]. Advances in Neural Information Processing Systems, 2022, 35: 7065-7076.

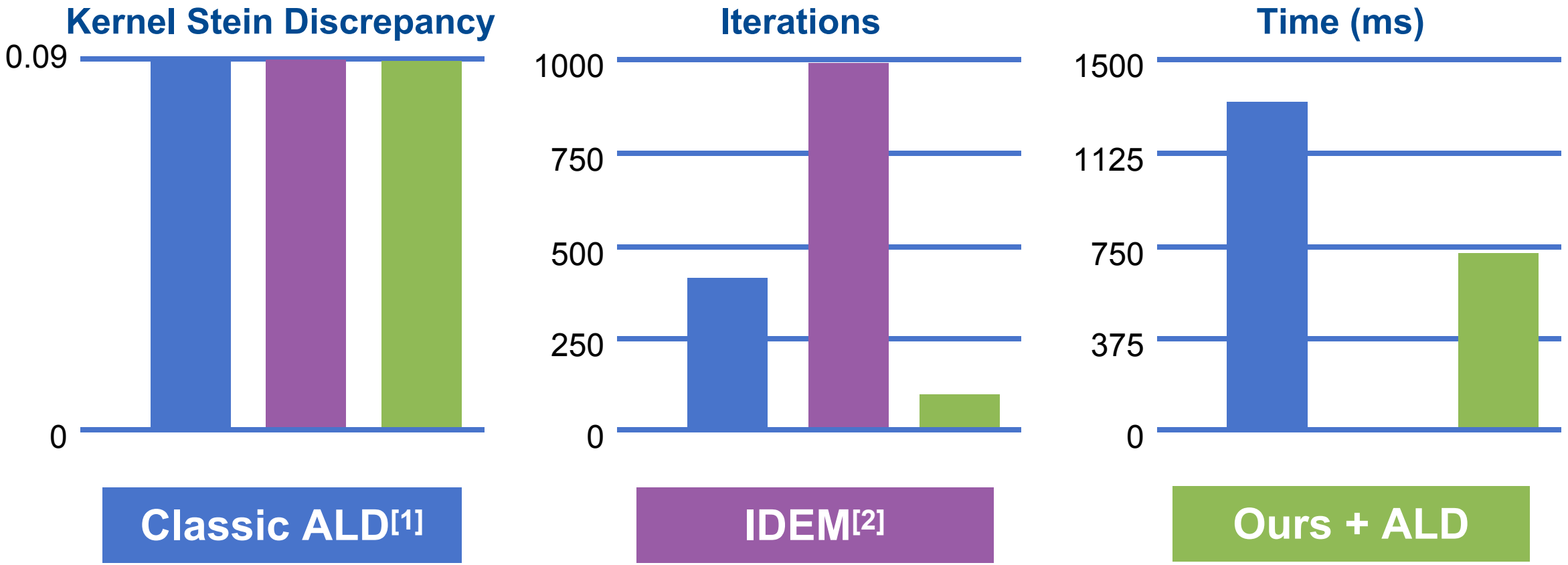
Problem 3 : Polynomial Root-Finding via HC



Same Success Rate, Less Iterations, Less Time

[1] Bates D J, Sommese A J, Hauenstein J D, et al. Numerically solving polynomial systems with Bertini[M]. Society for Industrial and Applied Mathematics, 2013.
[2] Zhang X, Dai Z, Xu W, et al. Simulator hc: Regression-based online simulation of starting problem-solution pairs for homotopy continuation in geometric vision[C]//Proceedings of the Computer Vision and Pattern Recognition Conference. 2025: 27103-27112.

Problem 4 : Sampling via ALD



Compariable Performance, Less Iterations, Less Time

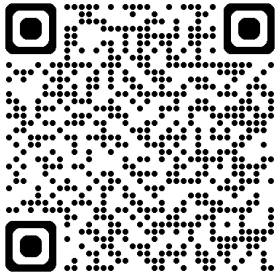
[1] Song Y, Sohl-Dickstein J, Kingma D P, et al. Score-based generative modeling through stochastic differential equations[J]. arXiv preprint arXiv:2011.13456, 2020.
[2] Akhound-Sadegh T, Rector-Brooks J, Bose A J, et al. Iterated denoising energy matching for sampling from boltzmann densities[J]. arXiv preprint arXiv:2402.06121, 2024.

Neural Predictor-Corrector: Solving Homotopy Problems with Reinforcement Learning

Same Performance

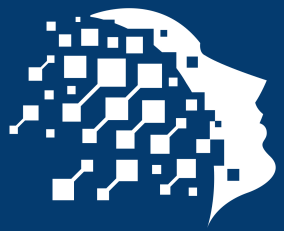
Less Iterations

Unified Framework



GitHub

https://github.com/maijiayao1/NPC



Thanks for watching