Machine Learning for Quantum Control

Barry C. Sanders
Seyed Shakib Vedaie, Archismita Dalal & Eduardo J. Páez

10 May 2019

Proc ITNG 2010:506; Proc ESANN 2016:327; *Neurocomputing* **268**:116 (2017)
PRL **104**:063603 (2010), **107**:233601 (2011), **110**:220501 (2013)
NJP **20**:113009 (2018); arXiv:1809.05525 (2018)

UNIVERSITY OF CALGARY NSERC CRSNG CIFAR CANADIAN INSTITUTE for ADVANCED RESEARCH Alberta Innovates Technology Futures New Journal of Physics The open access journal at the forefront of physics

Aim

Feasible control policies for quantum ($\mathscr{Q}$) technologies.

Claims

- Framework connecting control (C) & learning for both $\mathscr{Q}$ & classical ($\mathscr{C}$).
- A supervised-learning (SL) agent devises feasible control policies for phase estimation in adaptive $\mathscr{Q}$-enhanced metrology (A$\mathscr{Q}$EM).

Novelty

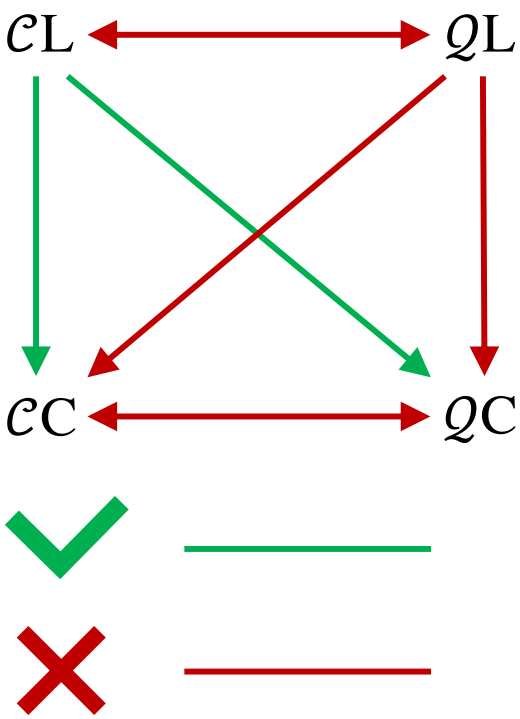
- Unification of $\mathscr{Q}$C and classical control ($\mathscr{C}$C).
- Method for casting $\mathscr{Q}$C as machine learning.

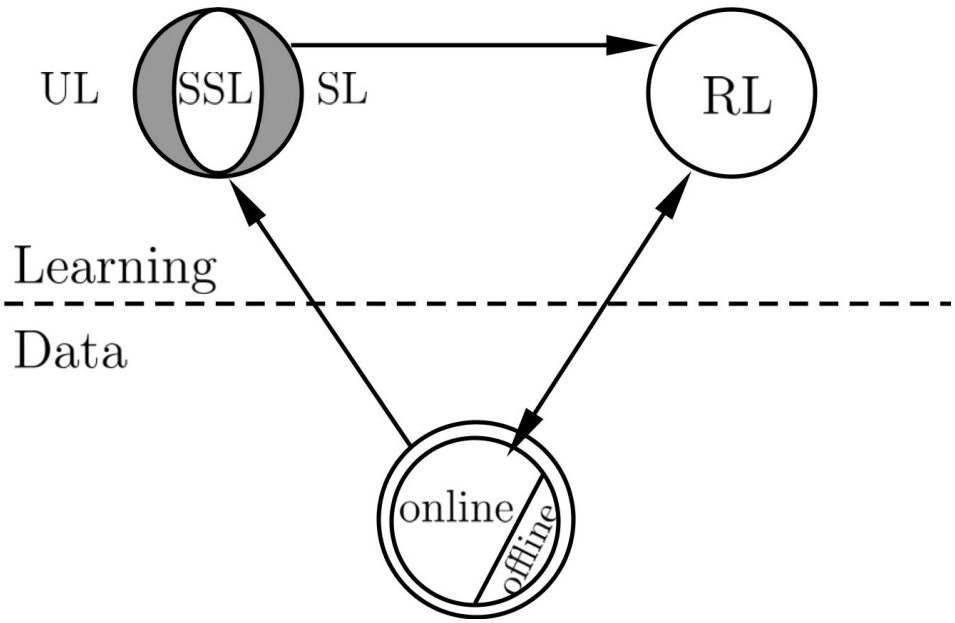
Importance

- Unconfuse.
- Enhance $\mathscr{Q}$C toolkit.

Learning

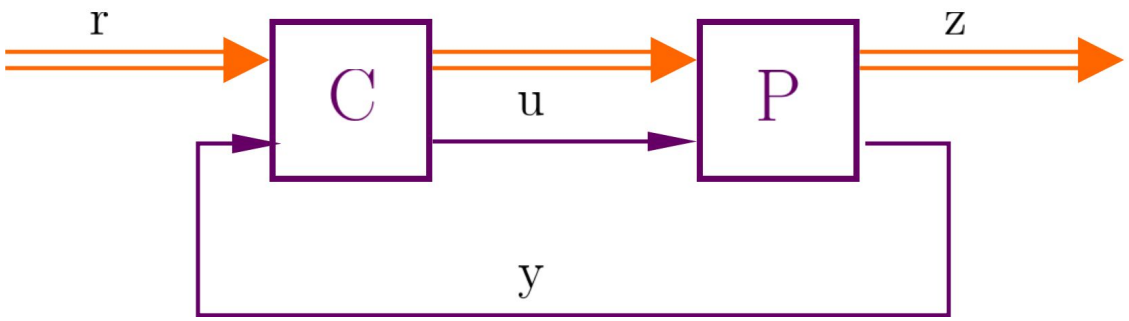
"An agent is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." — Tom Mitchell (1997) *Machine Learning*.

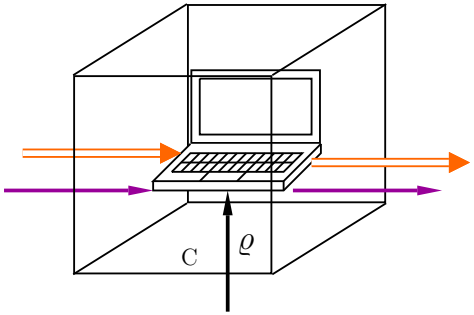




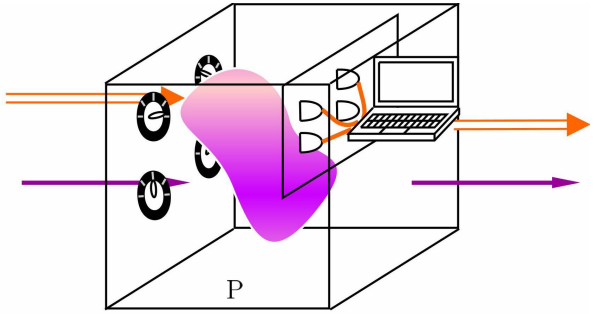
Unifying $\mathscr{Q}$C and $\mathscr{C}$C

Steer specific controllable degrees of freedom so plant dynamics yields approximately correct observations.





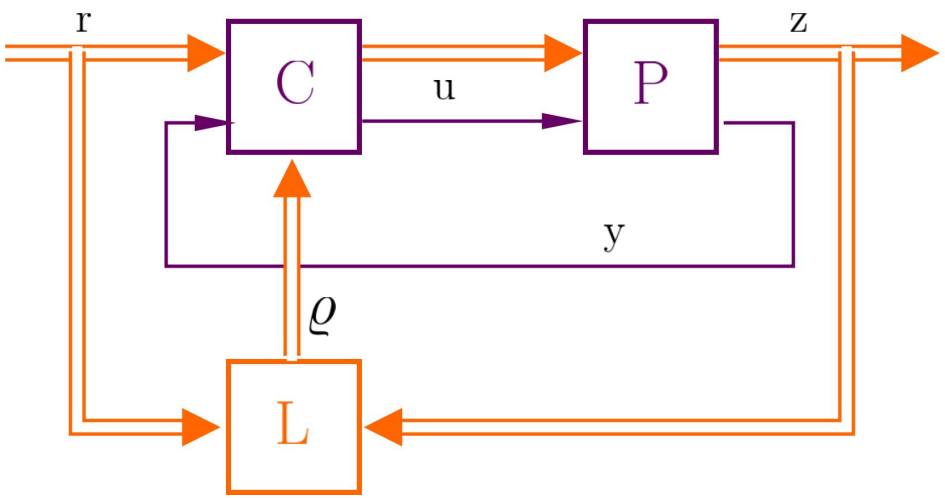
C: controller
ϱ: control policy



P: plant (system to be controlled+auxiliary)

Lewis and Yang (1997) *Basic Control Systems Engineering*

Learning for control



L: Learning agent ($\mathscr{C}$)

Learning enables a controller who is neither omniscient nor possesses a feasible alternative to execute the task successfully.

Fu (1986) doi.org/cgkfk2

A$\mathscr{Q}$EM

Simplest control task: U(1) transformation.

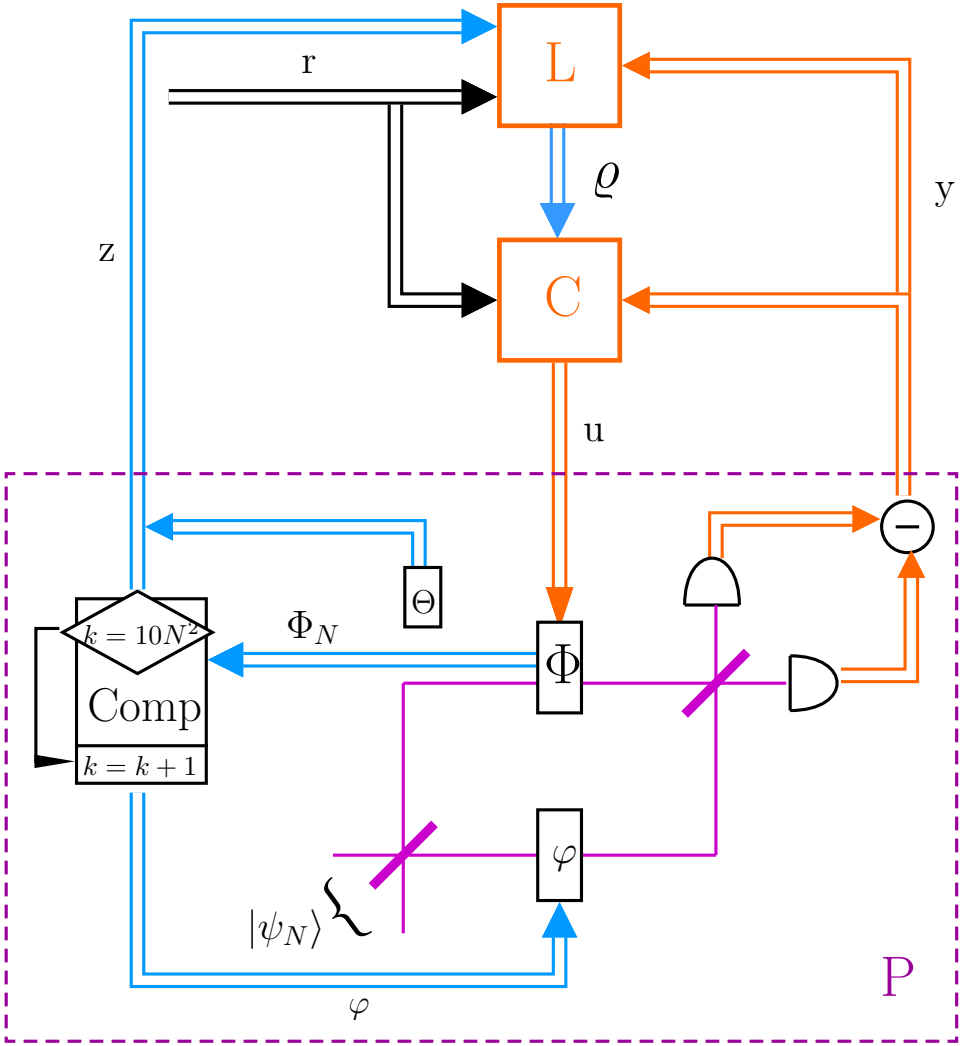
Task

Estimate unknown phase φ given N particles.

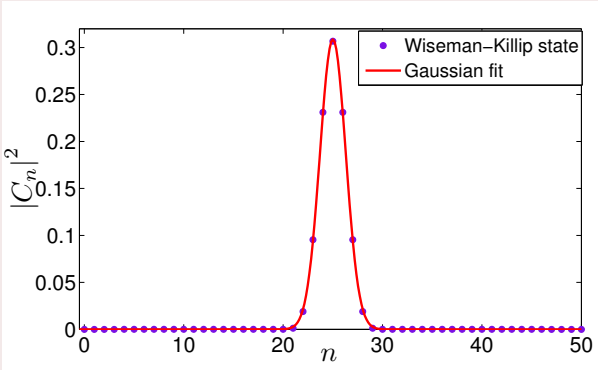
Uncertainty

$$\Delta\tilde{\varphi} \sim N^{-\wp}, \wp = \begin{cases} 1/2, & \text{Standard } \mathscr{Q} \text{ limit,} \\ 1, & \text{Heisenberg limit.} \end{cases}$$

SL for A$\mathscr{Q}$EM



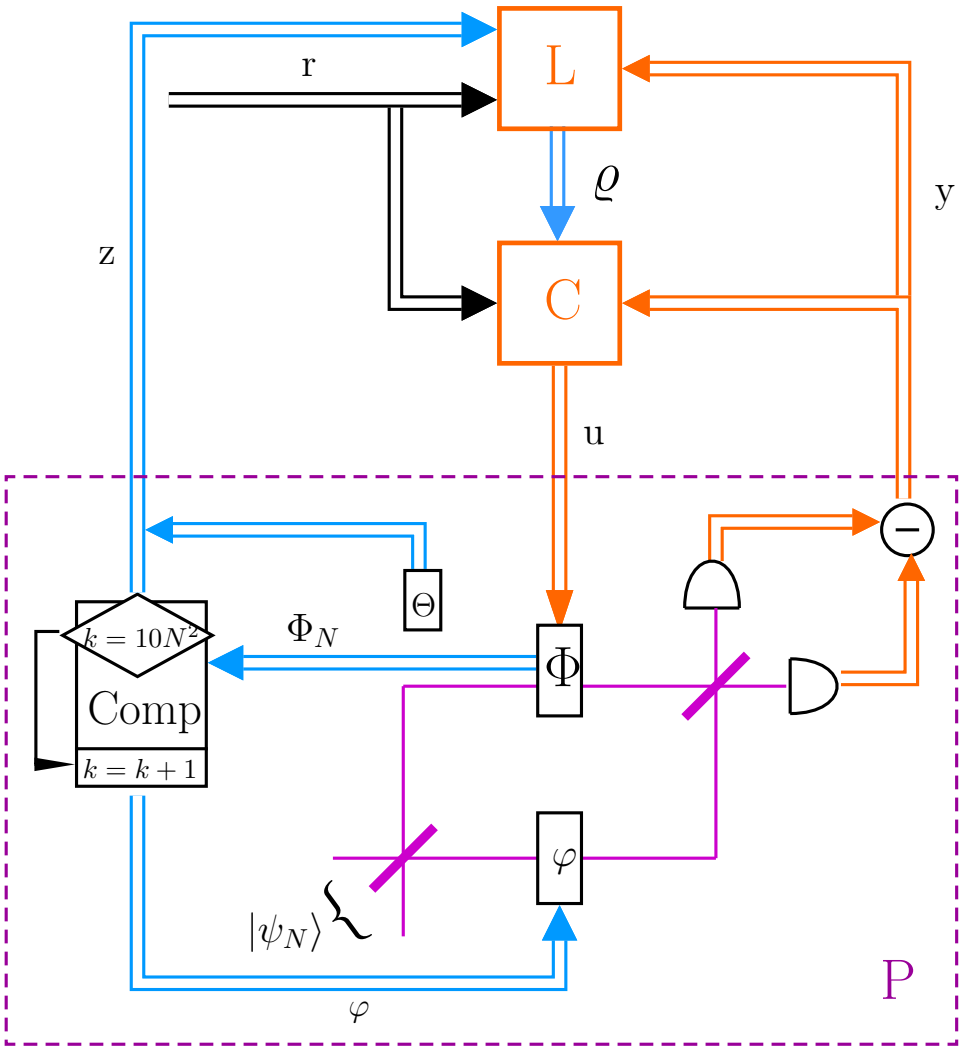
Input state and control



$$|\psi_N\rangle = \sum_{n=0}^{N} C_n \, |n_a, N - n_a\rangle$$

- Use differential evolution to search for feasible policies $\{\varrho\}$

SL for A$\mathscr{Q}$EM



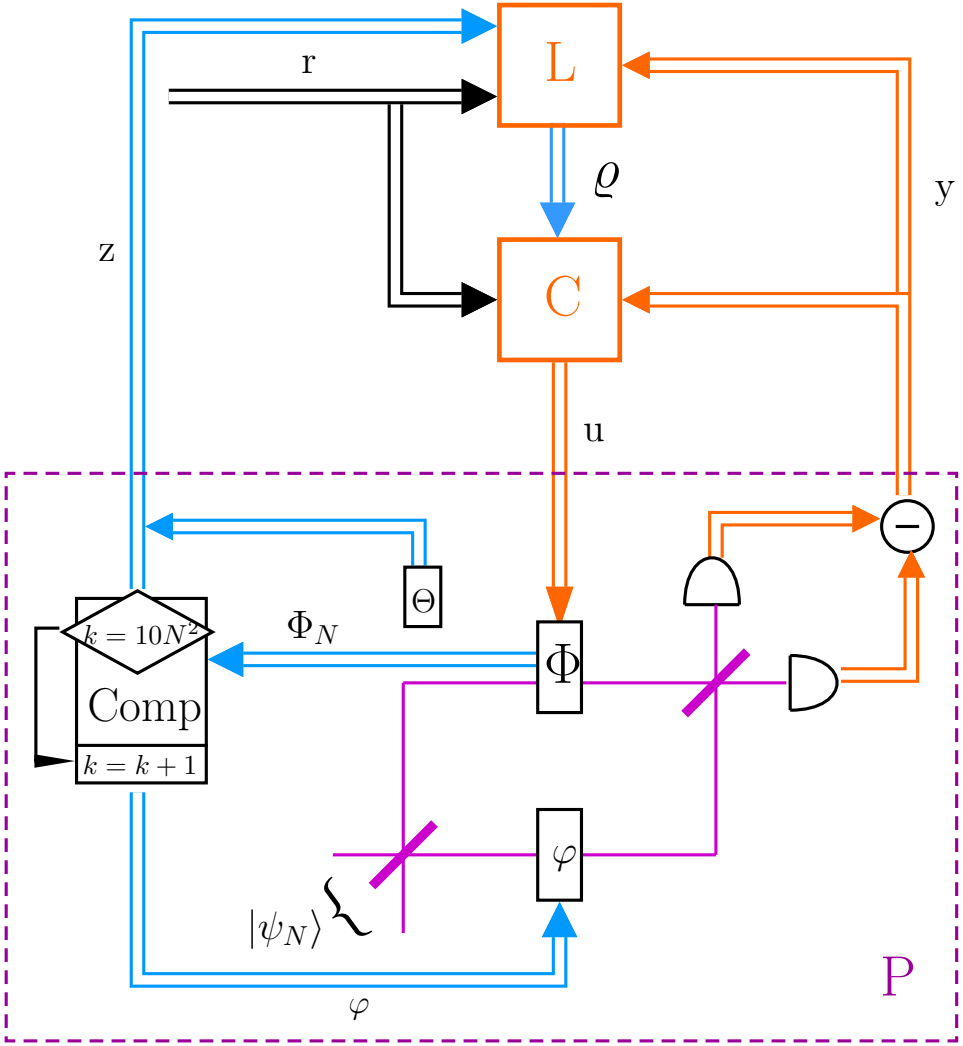
Training stage

- Feature vector $\boldsymbol{y} \in \{0,1\}^N$
- Label $\varphi \in \text{rand}\,[0, 2\pi)$
- Training set: $\{\boldsymbol{y}, \varphi\}$
- Hypothesis function $\Phi^{\varrho} : Y \to \{\varphi\} : \boldsymbol{y} \mapsto \varphi$
- Cost function

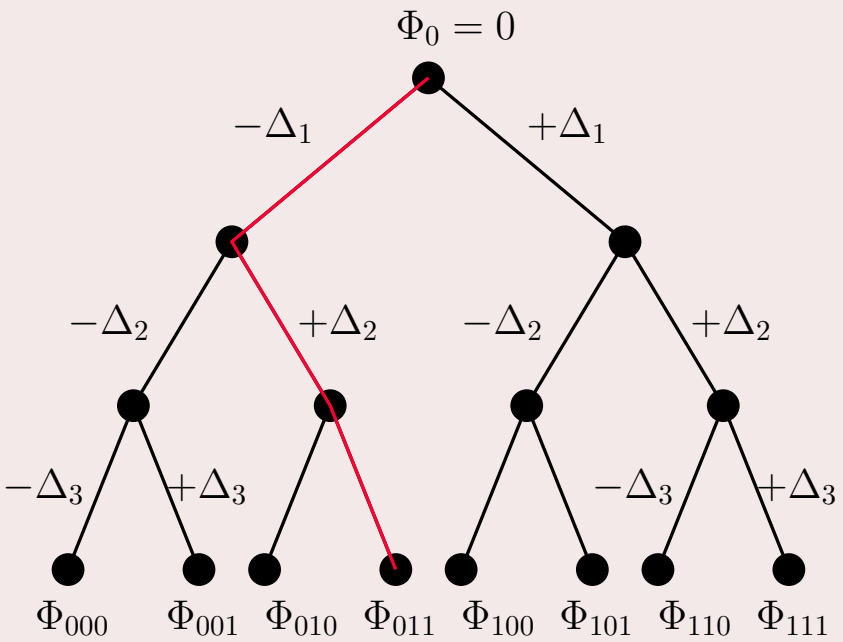
$$\mathrm{z} \leftarrow V_N^{\varrho} := (S_N^{\varrho})^{-2} - 1 \text{ for}$$

$$S_N^{\varrho} := \left| \sum_{k=1}^{10N^2} \frac{\exp \mathrm{i} \left(\varphi^{(k)} - (\Phi_N^{\varrho})^{(k)} \right)}{10N^2} \right|$$

SL for A$\mathscr{Q}$EM



Φ^{ϱ}: generalized log search



$$\Phi_0 = 0, y_m \in \{0, 1\}$$

$$\Phi_m = \Phi_{m-1} - (-1)^{y_m} \Delta_m$$

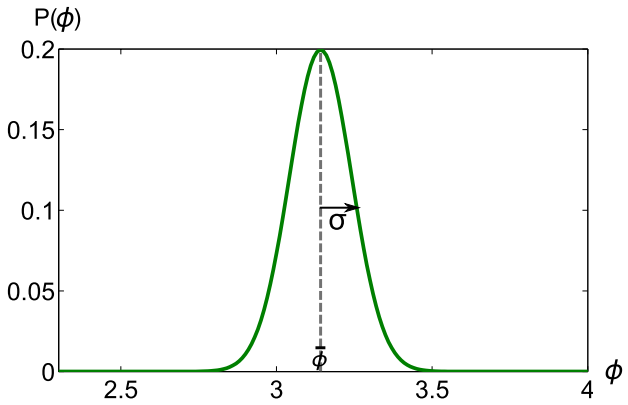
$$\varrho = (\Delta_1, \Delta_2, \ldots, \Delta_N) \in [0, 2\pi)^N$$

$$N = \{4, 5, \ldots, 100\}$$

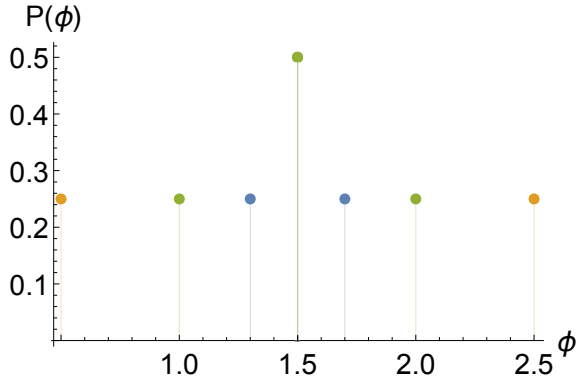
Phase-noise models

Symmetric distributions

Normal

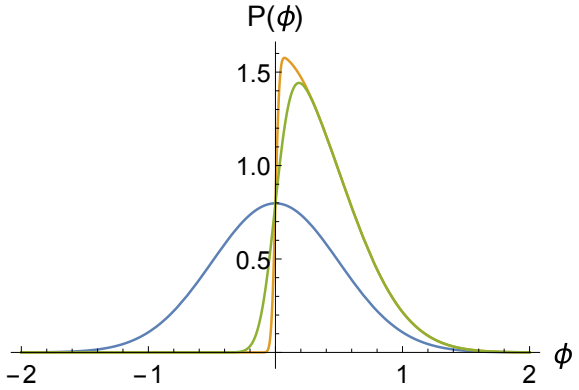


Random telegraph

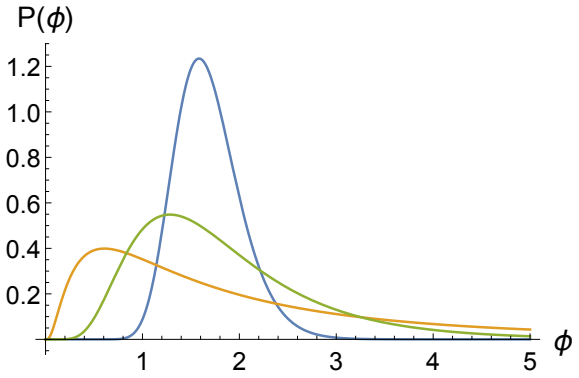


Asymmetric distributions

Skew-normal

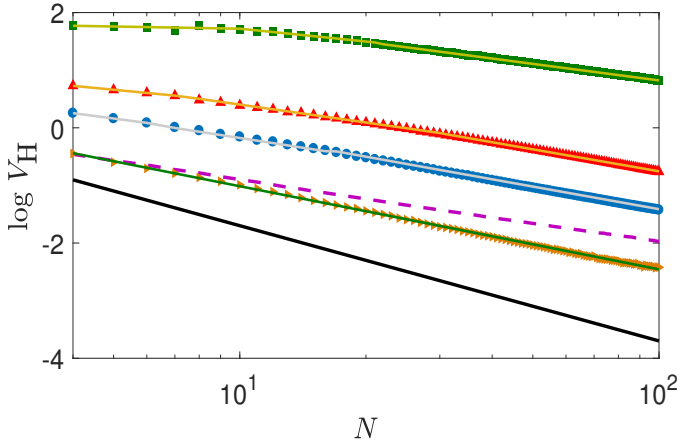


Log-normal

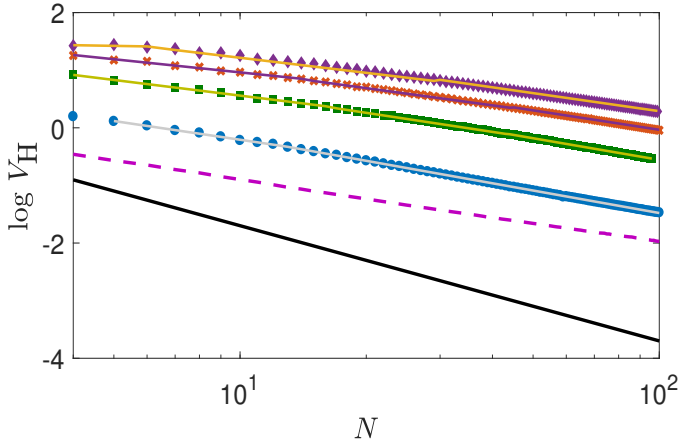


Results: SL for A$\mathscr{2}$EM

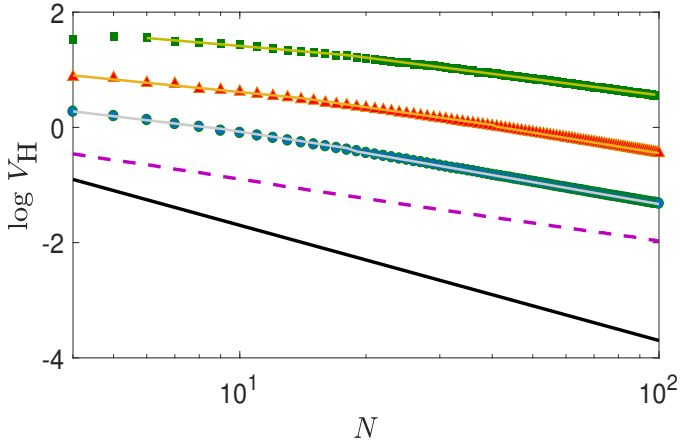
(a) normal-distribution noise



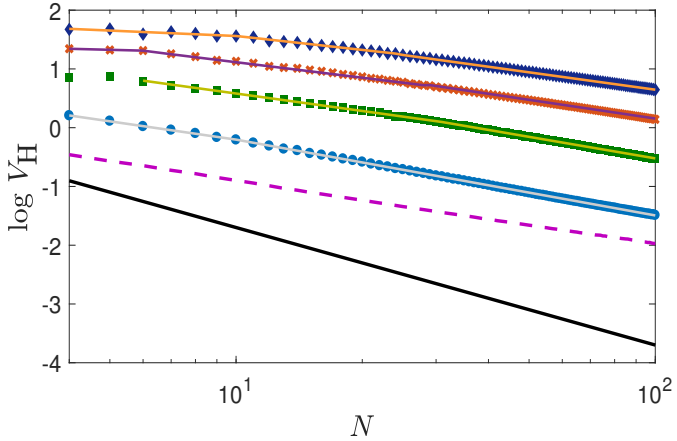
(c) skew-normal-distribution noise



(b) random telegraph noise



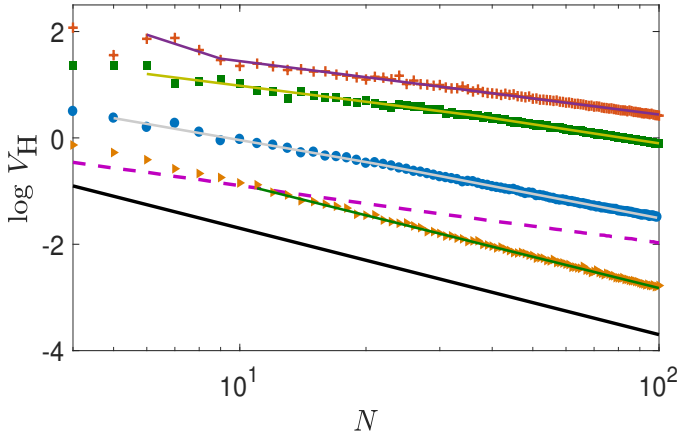
(d) log-normal-distribution noise



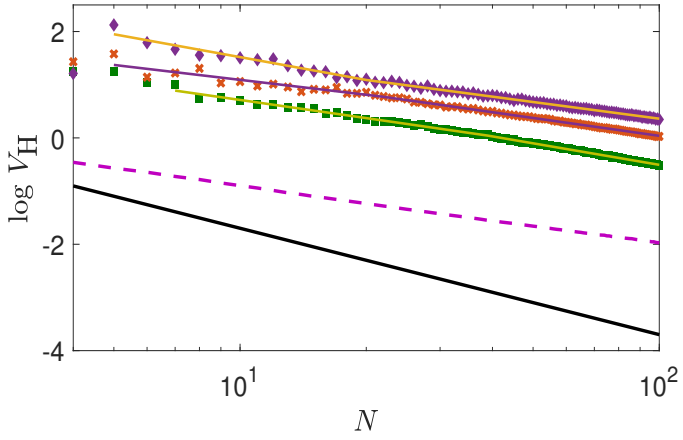
- for $V = 1$, △ for $V = 2$, ■ for $V = 3$, brown + for $V = 4$, brown x when $V = 5$, and purple ⋄ for $V = 7$.

Results: Bayesian feedback for A$\mathscr{2}$EM

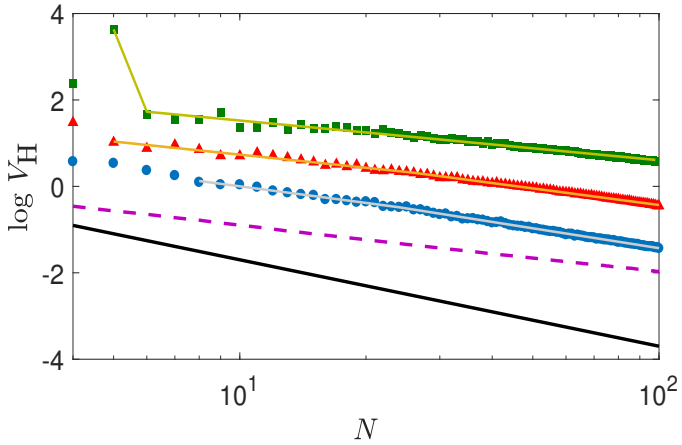
(a) normal-distribution noise



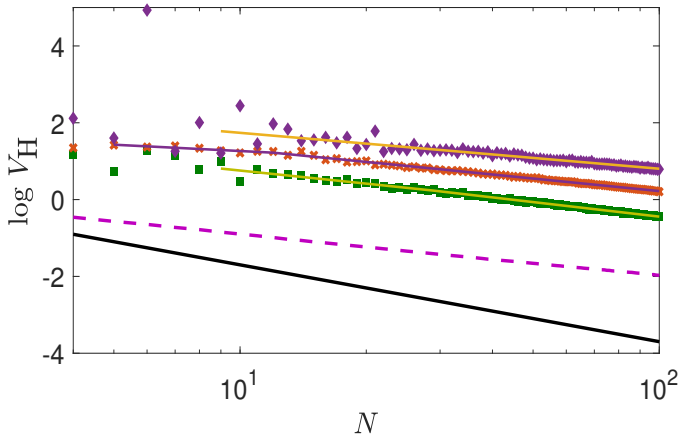
(c) skew-normal-distribution noise



(b) random telegraph noise



(d) log-normal-distribution noise



The blue $\circ$ are data when $V = 1$, the red triangles when $V = 2$, green $\square$ when $V = 3$ brown $+$ when $V = 4$, the brown crosses when $V = 5$, and purple $\diamond$ when $V = 7$.

Results: Power-law scaling

	V	γ	$2\wp_{\mathrm{SL}}$	$\overline{R^2}_{\mathrm{SL}}$	$2\wp_{\mathrm{B}}$	$\overline{R^2}_{\mathrm{B}}$
SQL			1		1	
HL			2		2	
No noise			1.459	0.9998	1.957	0.9993
Normal	1	0	1.302	0.9999	1.512	0.9985
	2	0	1.267	0.9999	–	–
	3	0	0.954	0.9992	1.190	0.9997
	4	0	–	–	1.004	0.9948
Random telegraph	1	0	1.266	0.9999	1.526	0.9991
	2	0	1.186	0.9997	1.277	0.9967
	3	0	0.935	0.9993	0.919	0.9892
Skew-normal	1	0.8509	1.296	0.9999	–	–
	3	0.8509	1.246	0.9999	1.343	0.9987
	5	0.8509	1.118	0.9998	1.116	0.9927
	7	0.8509	1.039	0.9996	1.041	0.9964
Log-normal	1	0.8509	1.290	0.9999	–	–
	3	0.8509	1.217	0.9998	1.258	0.9919
	5	0.8509	1.058	0.9997	1.086	0.9961
	7	0.8509	0.981	0.9994	0.9209	0.7965

Complexity	SL	BF
Design time	$O(N^6)$	–
Policy space	$O(N)$	$O(N^2)$
Implementation time	$O(N)$	$O(N^3)$

- SL policies delivers $\wp_{\mathrm{L}} > 1/2$, but not better than Bayesian (model-based) policies.
- Learned policies are computationally cheaper than Bayesian method.