Quantum Algorithms for Multilevel Monte Carlo Methods

Yassine Hamoudi

CNRS, LaBRI, Université de Bordeaux

Based on joint work with J. Blanchet, M. Szegedy, G. Wang

arXiv:2502.05094

Quantum Information & Computation @ LaBRI

Home Members Seminar Open positions Contact

LABORATOIRE BORDELAIS DE RECHERCHE EN INFORMATIQUE

We are a research group in quantum computing, formally created in 2024 and located in Bordeaux (France). Our members are hosted at LaBRI in Université de Bordeaux. We are also part of the CNRS research networks in quantum information (GT IQ) and in quantum technologies (GDR TeQ).

Our areas of research and interest include:

Quantum Algorithms and Computational Speedups

- Optimization and Monte Carlo methods
- Combinatorial algorithms
- Distributed algorithms
- Simulation of quantum systems

Quantum Information and Complexity Theory

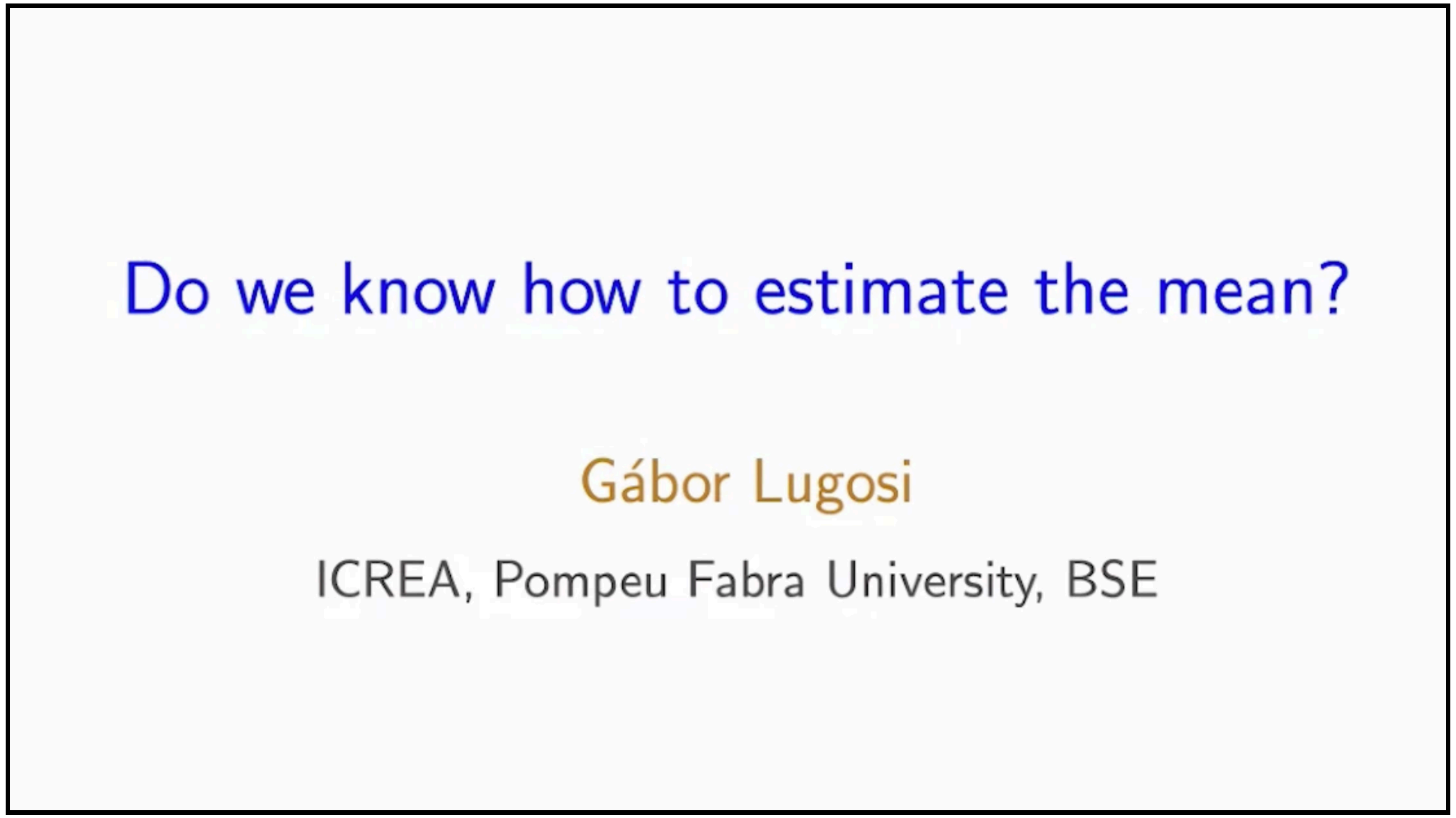
- Classical and quantum query complexity
- Limitations of near-term quantum computers
- Simulation of quantum circuits
- Quantum graphs

Quantum Computing and Fundamental Physics

- Quantum simulation of strongly interacting fermionic systems
- Quantum computing and quantum field theory
- Holographic complexity

https://quantique.labri.fr/

Invited talk (Breiman Lecture) at NeurIPS 2021



“Despite its long history, the subject has attracted a flurry of renewed activity. Motivated by applications in machine learning and data science, the problem has been viewed from new angles both from statistical and computational points of view.”

Estimating an expectation

- Fundamental task for extracting classical information from quantum systems

Estimating an expectation

- Fundamental task for extracting classical information from quantum systems

- Entanglement allows for better accuracy in the estimation (Heisenberg vs shot-noise limits)

Estimating an expectation

- Fundamental task for extracting classical information from quantum systems

- Entanglement allows for better accuracy in the estimation (Heisenberg vs shot-noise limits)

- Applications to algorithmic speedups: statistical and computational advantages using faster quantum estimators

Examples of expectation values

Expectation value of an observable

$$\langle \psi | O | \psi \rangle$$

Expected value of a stochastic process

$$E[f(X)]$$

Shadow tomography

$$|\psi\rangle^{\otimes k} \to 01101\ldots \to \{\langle \psi | O_i | \psi \rangle\}_i$$

Partition function of a Hamiltonian

$$Z(\beta) = \mathrm{Tr}\left(e^{-\beta H}\right)$$

Expected value of a stochastic process

Stochastic process
generating a
random variable X

$

Statistical estimator
computing the
expectation $E[f(X)]$

$\rightarrow (1 \pm \epsilon)\, E[f(X)]$

Expected value of a stochastic process

Stochastic process generating a random variable X

$

Statistical estimator computing the expectation $E[f(X)]$ $\rightarrow (1 \pm \epsilon)\, E[f(X)]$

Classical algorithms

$$\sim \mathrm{Var}(f(X))/\epsilon^2 \times \$$$

(standard quantum limit)

Quantum algorithms

$$\sim \sqrt{\mathrm{Var}(f(X))}/\epsilon \times \$$$

(Heisenberg limit)

[Hamoudi'2021] [Kothari, O'Donnell'2023]

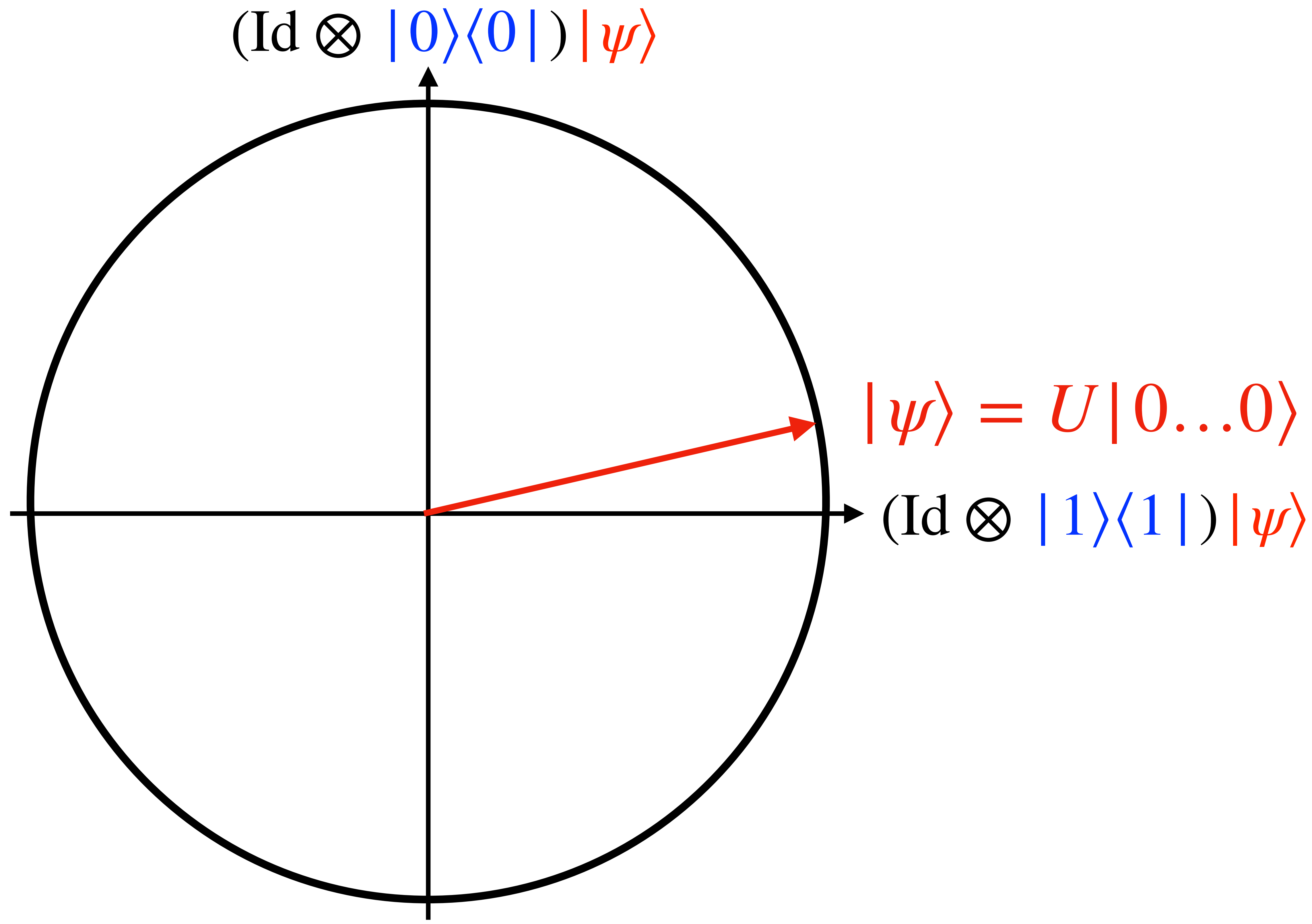
The case $f(X) \in \{0,1\}$

The case $f(X) \in \{0,1\}$

Run the stochastic process in superposition: $U|0\ldots0\rangle \propto \sum_{P\text{ : computation path}} \overset{\$}{|P\rangle} \overset{\in \{0,1\}}{\overset{\downarrow}{|f(X(P))\rangle}}$

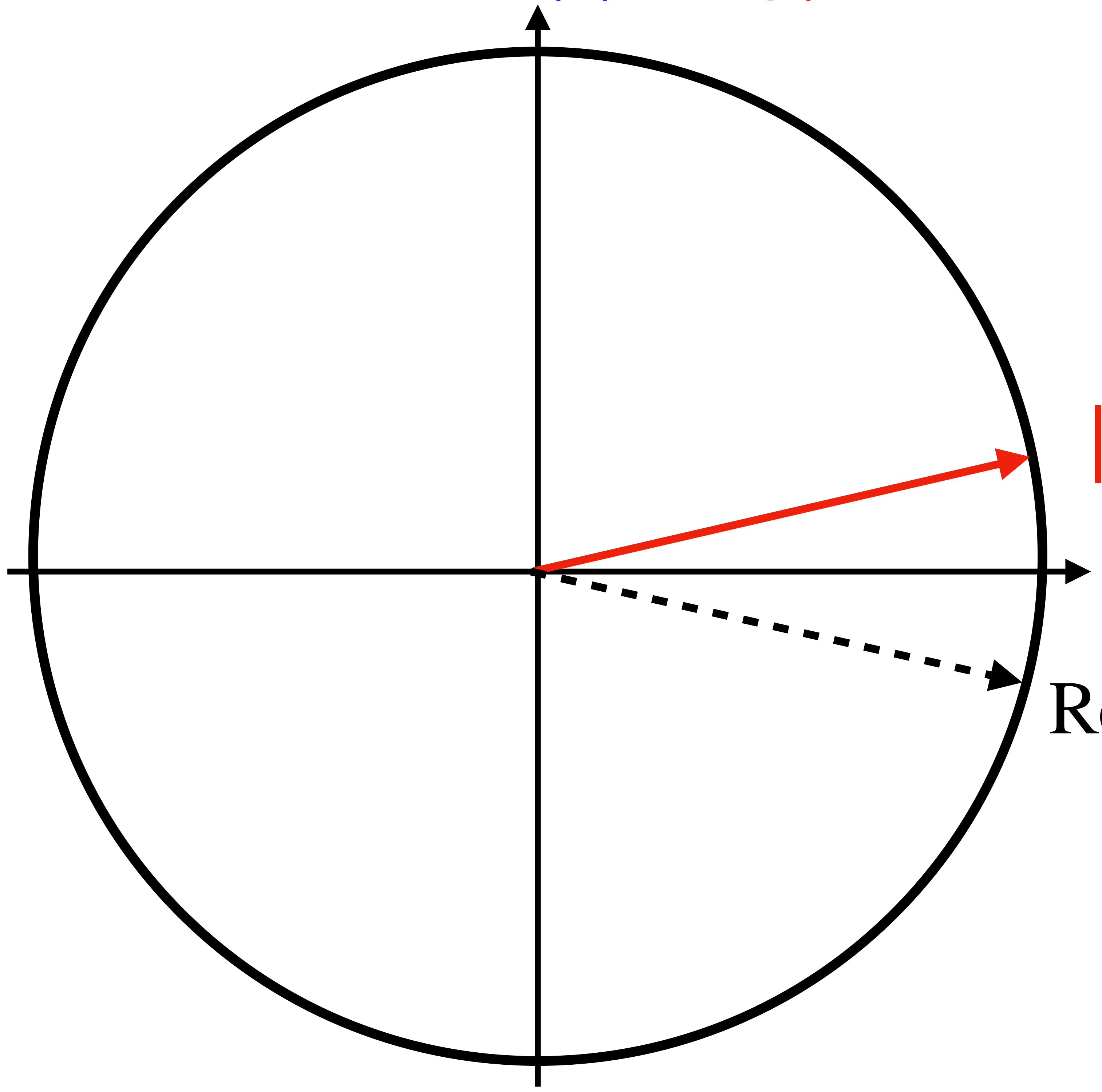
The case $f(X) \in \{0,1\}$

Run the stochastic process in superposition: $U|0\ldots0\rangle \propto \sum_{P\,:\,\text{computation path}} \overset{\$}{|P\rangle}\,|\overset{\in\{0,1\}}{f(X(P))}\rangle$



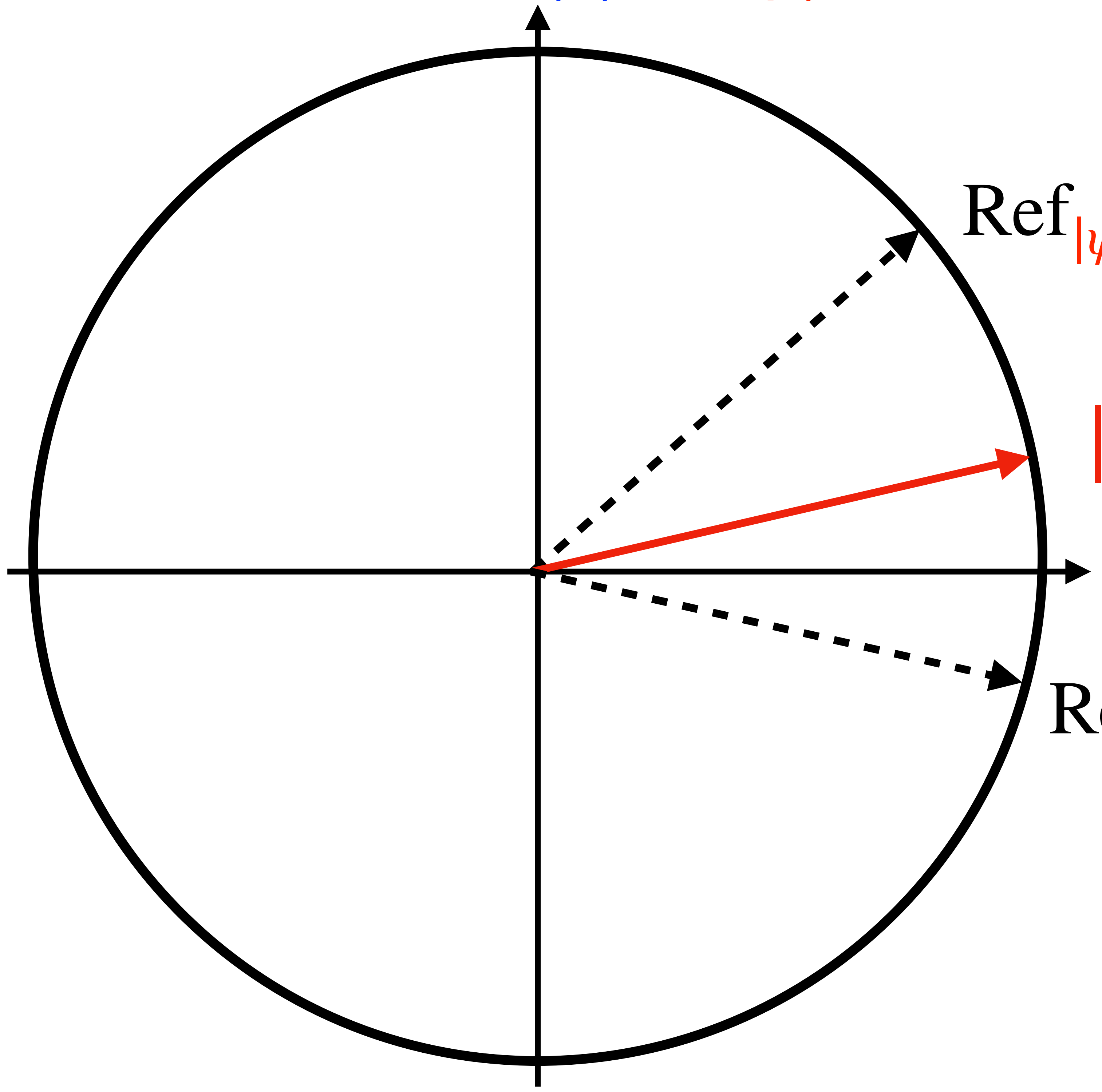
The case $f(X) \in \{0,1\}$

Run the stochastic process in superposition: $U|0\ldots0\rangle \propto \sum_{\text{P : computation path}} \overset{\$}{} \, |P\rangle \, |\overset{\in \{0,1\}}{\overset{\downarrow}{f(X(P))}}\rangle$



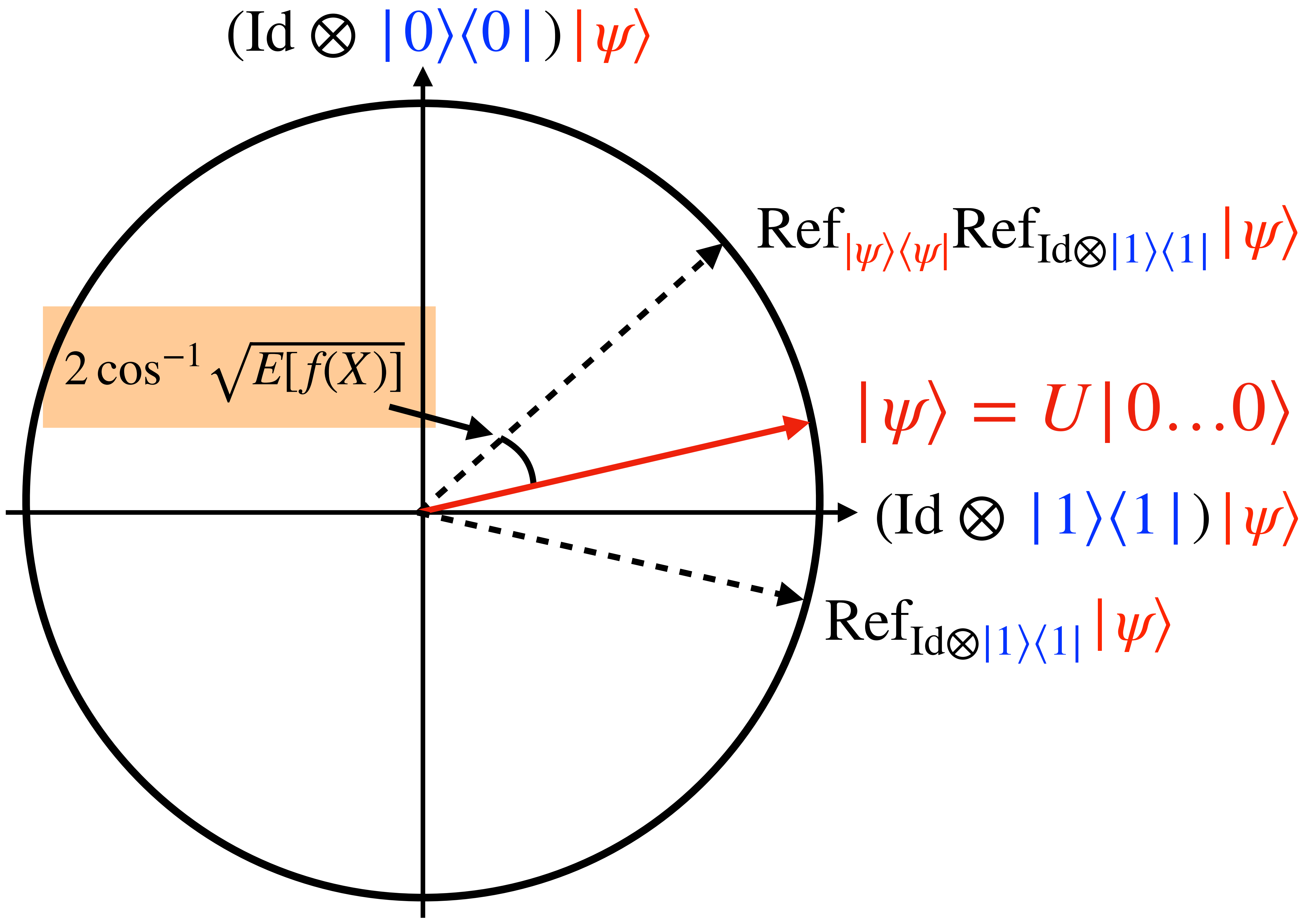
The case $f(X) \in \{0,1\}$

Run the stochastic process in superposition: $U|0\ldots0\rangle \propto \sum_{P\text{ : computation path}} \overset{\$}{} |P\rangle|\underbrace{f(X(P))}_{}\rangle$, $f(X(P)) \in \{0,1\}$



The case $f(X) \in \{0,1\}$

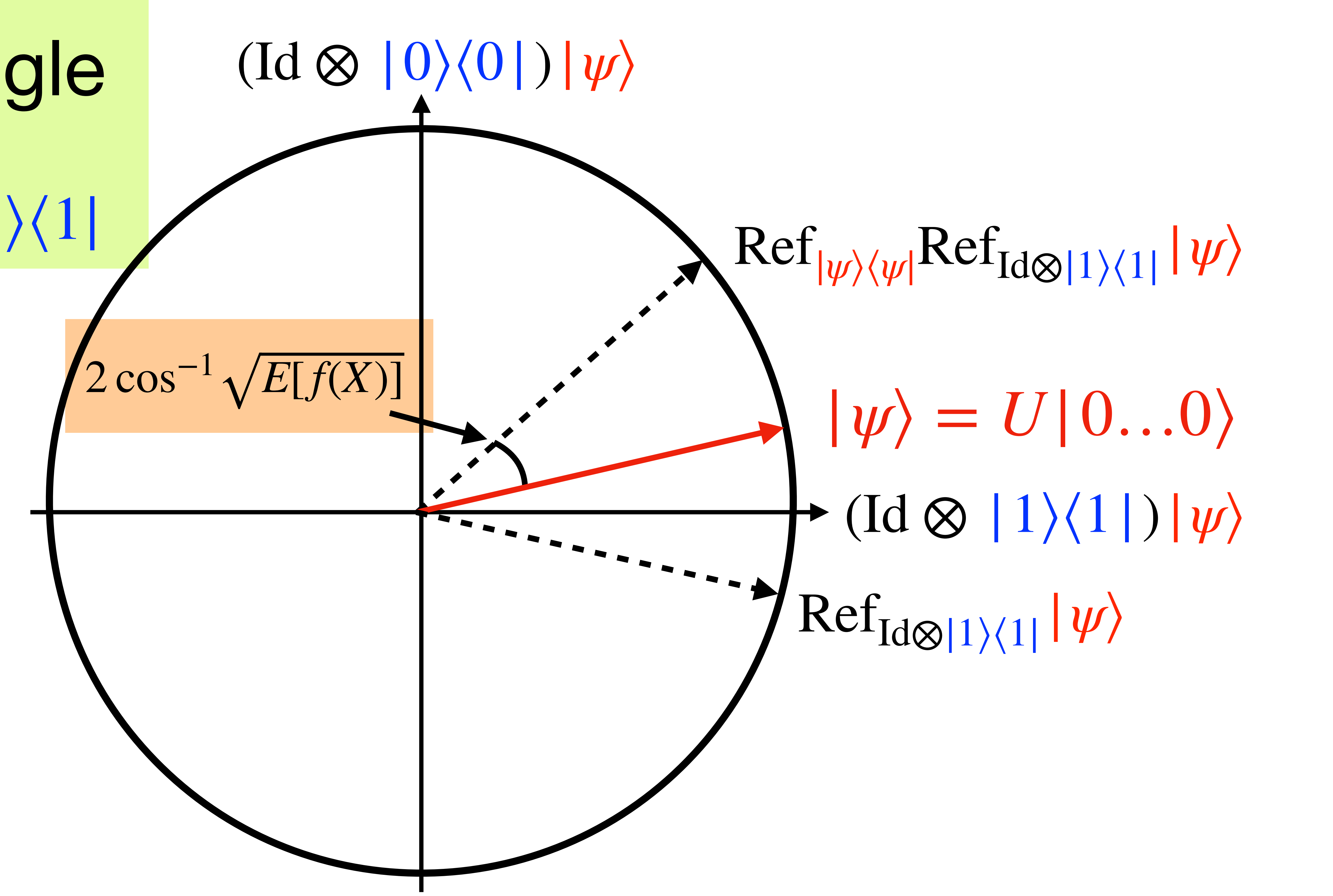
Run the stochastic process in superposition: $U|0\ldots0\rangle \propto \sum_{P\text{ : computation path}} \overset{\$}{|P\rangle} \underbrace{|f(X(P))\rangle}_{}^{\in\{0,1\}}$



The case $f(X) \in \{0,1\}$

Run the stochastic process in superposition: $U|0\ldots0\rangle \propto \sum_{P\,:\,\text{computation path}} \$\;|P\rangle|\underbrace{f(X(P))}_{\in\{0,1\}}\rangle$

$E[f(X)]$ is encoded into the rotation angle of $R = U \cdot \mathrm{Ref}_{|0\ldots0\rangle\langle0\ldots0|} \cdot U^\dagger \cdot \mathrm{Ref}_{\mathrm{Id}\otimes|1\rangle\langle1|}$

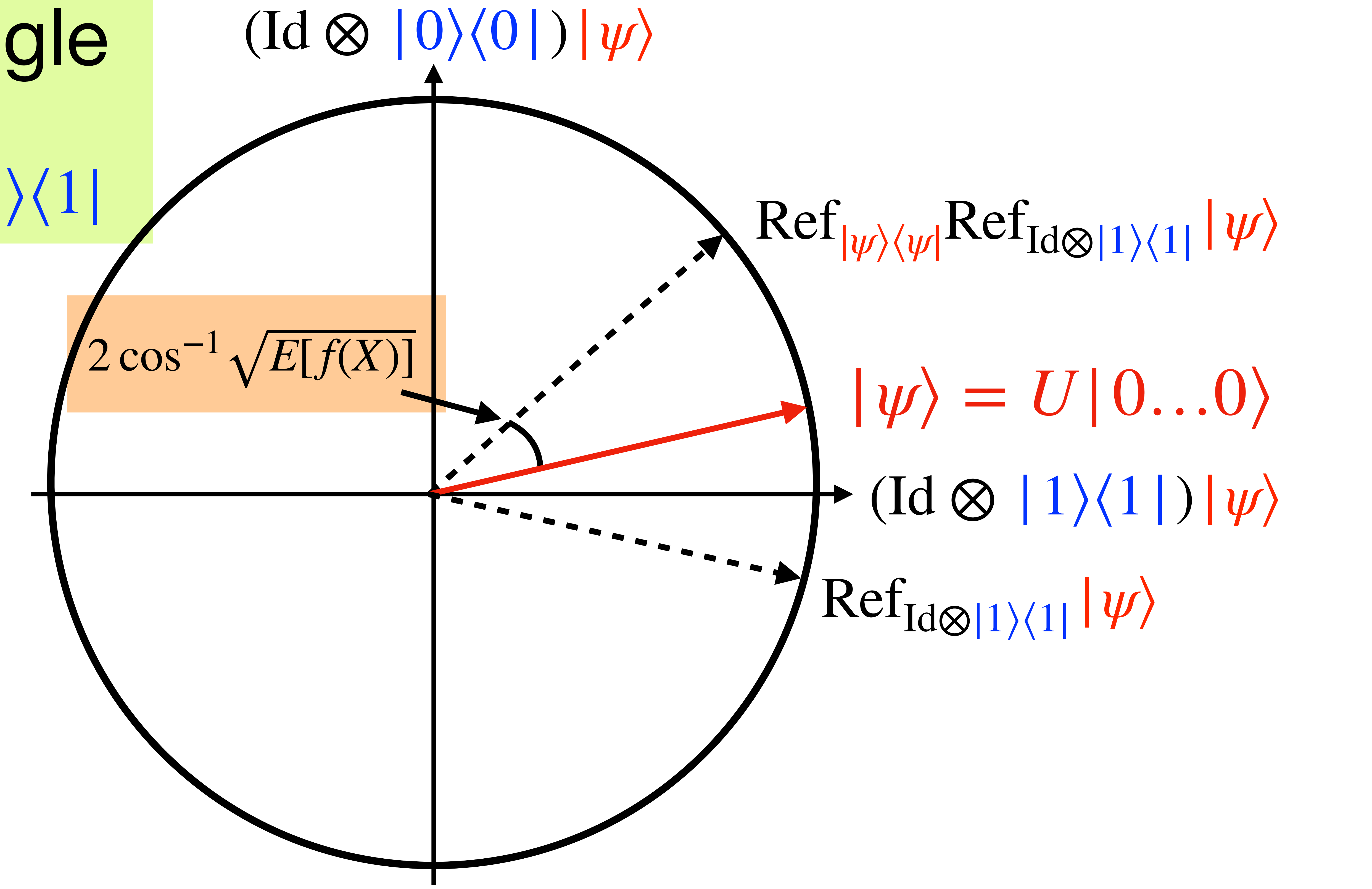


The case $f(X) \in \{0,1\}$

Run the stochastic process in superposition: $U|0\ldots0\rangle \propto \sum_{P\text{ : computation path}} \overset{\$}{|P\rangle} \, |\overset{\in\{0,1\}}{f(X(P))}\rangle$

$E[f(X)]$ is encoded into the rotation angle of $R = U \cdot \mathrm{Ref}_{|0\ldots0\rangle\langle 0\ldots0|} \cdot U^\dagger \cdot \mathrm{Ref}_{\mathrm{Id}\otimes|1\rangle\langle1|}$

Quantum Phase Estimation on R retrieves $(1 \pm \epsilon)\, E[f(X)]$ after $\sim 1/\epsilon$ executions of U



Expected value of a stochastic process

Classical algorithms	Quantum algorithms
$\sim \mathrm{Var}(f(X))/\epsilon^2 \times \$$	$\sim \sqrt{\mathrm{Var}(f(X))}/\epsilon \times \$$
(standard quantum limit)	(Heisenberg limit)

- In most applications, the cost $ of the generating process can be fine-tuned (tradeoff variance $\mathrm{Var}(f(X))$ / generating cost $)
- Does the apparent **quadratic speedup** survives when taking all costs into account (end-to-end analysis)?

MLMC applied to:

Stochastic Differential Equations

Stochastic differential equations

Estimate $(1 \pm \epsilon)\, E[f(X)]$ where X is solution (at time $t = 1$) to the SDE:

$$\mathrm{d}X = \mu(t, X)\,\mathrm{d}t + \sigma(t, X)\,\mathrm{d}W(t)$$

Drift coefficient

Diffusion coefficient

Standard Brownian motion

Stochastic differential equations

Estimate $(1 \pm \epsilon)\, E[f(X)]$ where X is solution (at time $t = 1$) to the SDE:

$$\mathrm{d}X = \mu(t, X)\,\mathrm{d}t + \sigma(t, X)\,\mathrm{d}W(t)$$

Drift coefficient

Diffusion coefficient

Standard Brownian motion

How to generate samples from X ?

Stochastic differential equations

Estimate $(1 \pm \epsilon)\, E[f(X)]$ where X is solution (at time $t = 1$) to the SDE:

$$\mathrm{d}X = \underbrace{\mu(t, X)}_{\text{Drift coefficient}}\,\mathrm{d}t + \underbrace{\sigma(t, X)}_{\text{Diffusion coefficient}}\,\mathrm{d}\underbrace{W(t)}_{\text{Standard Brownian motion}}$$

How to generate samples from X ?

- analytic solutions (Ex: Geometric Brownian motion: $\mu(t, X) = \mu X$ and $\sigma(t, X) = \sigma X$

$\rightarrow X(t) = X(0)\exp((\mu - \sigma^2/2)t + \sigma W(t))$

Stochastic differential equations

Estimate $(1 \pm \epsilon)\, E[f(X)]$ where X is solution (at time $t = 1$) to the SDE:

$$\mathrm{d}X = \mu(t, X)\,\mathrm{d}t + \sigma(t, X)\,\mathrm{d}W(t)$$

Drift coefficient

Diffusion coefficient

Standard Brownian motion

How to generate samples from X ?

- analytic solutions (Ex: Geometric Brownian motion: $\mu(t, X) = \mu X$ and $\sigma(t, X) = \sigma X$

$\rightarrow X(t) = X(0)\exp((\mu - \sigma^2/2)t + \sigma W(t))$)

- discretization schemes (Ex: Euler method: $X_{t+\delta} = X_t + \mu(t, X_t)\delta + \sigma(t, X_t)\Delta W_t$)

Stochastic differential equations

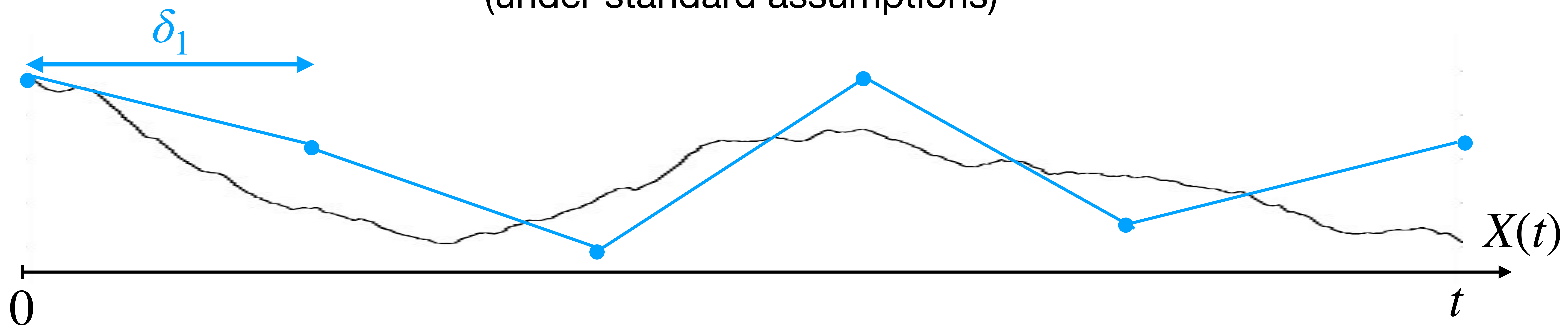
Euler discretization

(under standard assumptions)

Stochastic differential equations

Euler discretization

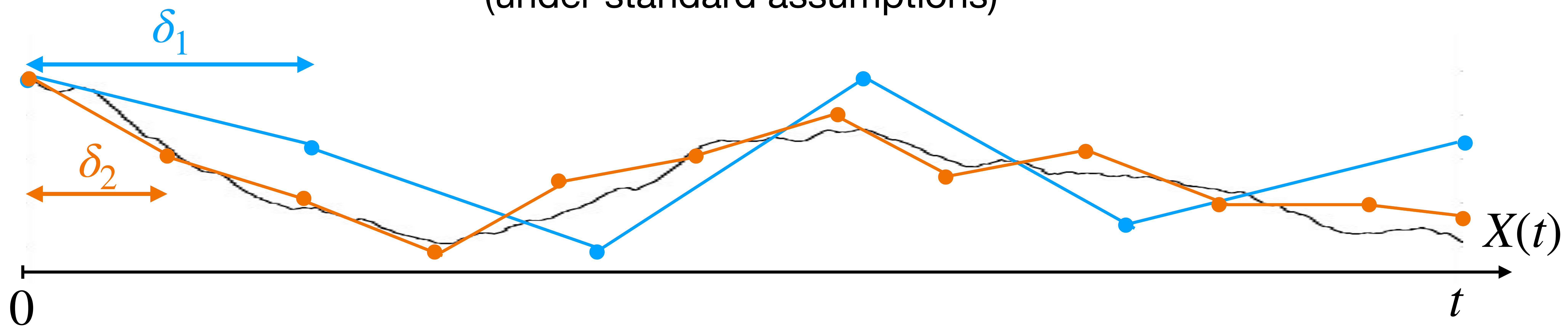
(under standard assumptions)



Stochastic differential equations

Euler discretization

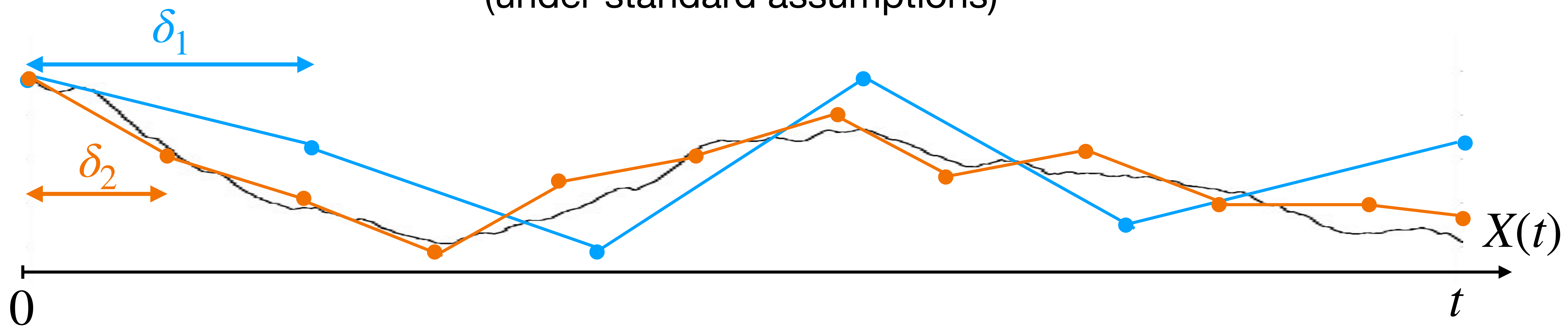
(under standard assumptions)



Stochastic differential equations

Euler discretization

(under standard assumptions)



Classical (naive)

Step size: $\delta \sim \epsilon$

Generation cost: **\$** $\sim 1/\delta$

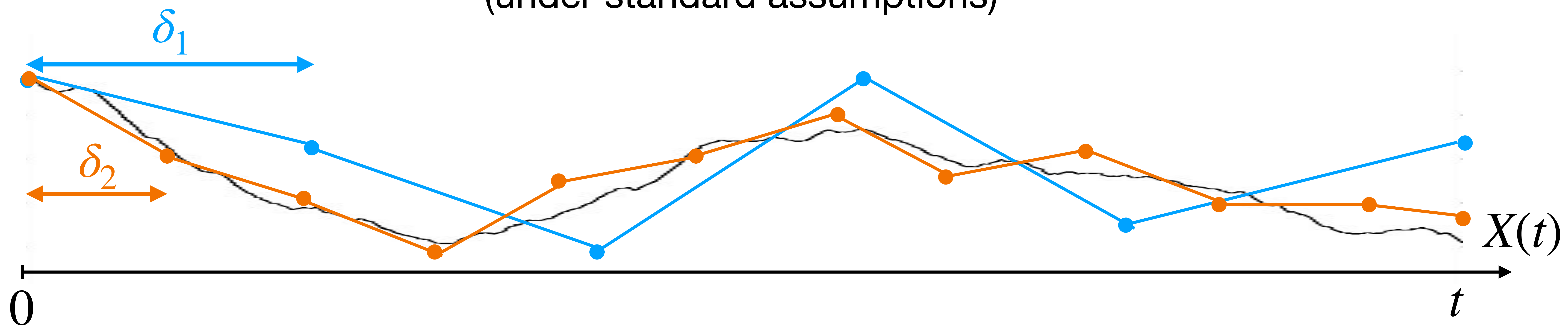
Overall cost: $\sim 1/\epsilon^2 \times$ **\$** $\sim 1/\epsilon^3$

Cubic scaling!

Stochastic differential equations

Euler discretization

(under standard assumptions)



Classical (naive)

Step size: $\delta \sim \epsilon$

Generation cost: $\$ \sim 1/\delta$

Overall cost: $\sim 1/\epsilon^2 \times \$ \sim 1/\epsilon^3$

Cubic scaling!

Quantum (naive)

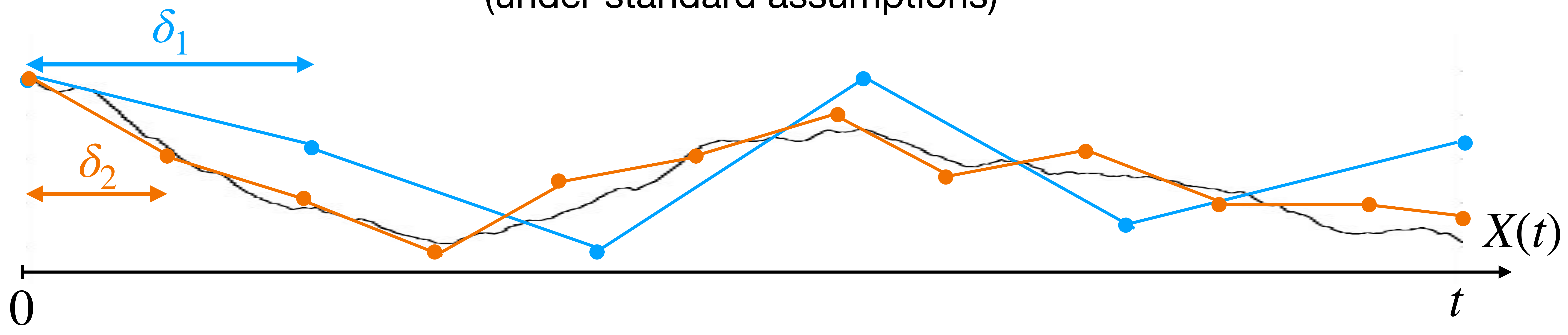
Same

Overall cost: $\sim 1/\epsilon \times \$ \sim 1/\epsilon^2$

Stochastic differential equations

Euler discretization

(under standard assumptions)



Classical (naive)

Step size: $\delta \sim \epsilon$

Generation cost: $ \sim 1/\delta$

Overall cost: $\sim 1/\epsilon^2 \times \$ \sim 1/\epsilon^3$

Cubic scaling!

Quantum (naive)

Same

Overall cost: $\sim 1/\epsilon \times \$ \sim 1/\epsilon^2$

Classical (MLMC)

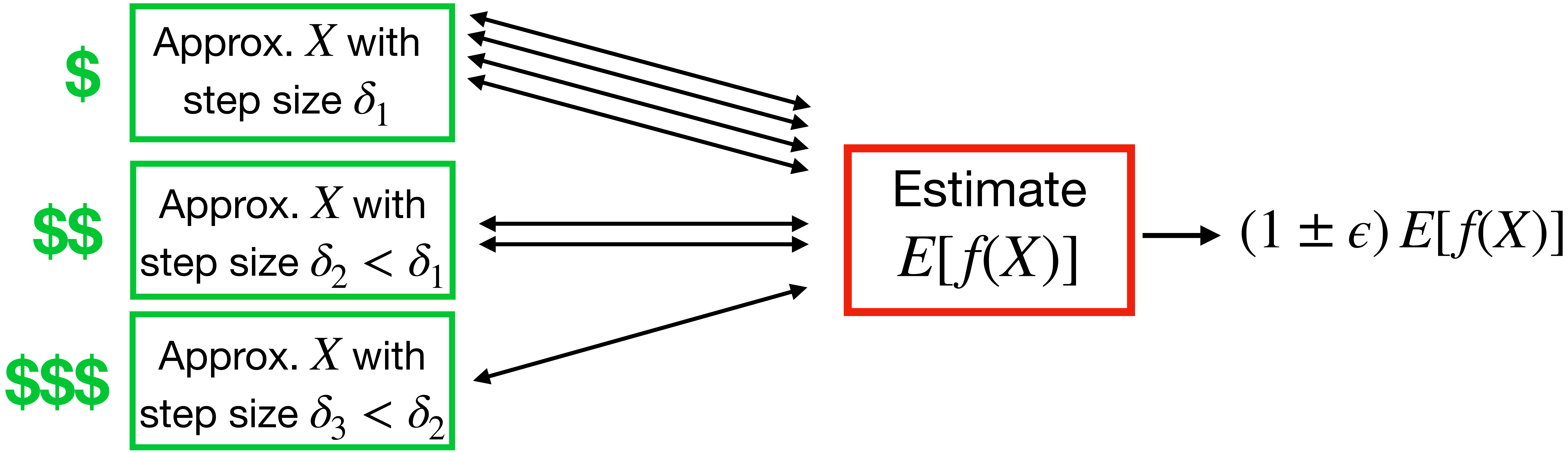
?

Overall cost: $\sim 1/\epsilon^2$

Same quadratic scaling!

Stochastic differential equations

(Classical) Multilevel Monte Carlo (MLMC) Method



Overall balanced cost: $\sim 1/\epsilon^2$ [Giles'2008]

Stochastic differential equations

MLMC with Euler discretization

$$(1 \pm \epsilon)\, E[f(X)]$$

(under standard assumptions)

Classical (naive)	Quantum (naive)	Classical (MLMC)	Quantum (MLMC)
$\sim 1/\epsilon^3$	$\sim 1/\epsilon^2$	$\sim 1/\epsilon^2$	$\sim 1/\epsilon^{1.5}$
		[Giles'2008]	[An et al.'2021]
			Sub-quadratic speedup

Stochastic differential equations

MLMC with Euler discretization

$$(1 \pm \epsilon)\, E[f(X)]$$

(under standard assumptions)

Classical (naive)	Quantum (naive)	Classical (MLMC)	Quantum (MLMC)
$\sim 1/\epsilon^3$	$\sim 1/\epsilon^2$	$\sim 1/\epsilon^2$	$\sim 1/\epsilon^{1.5}$
		[Giles'2008]	[An et al.'2021]
			Sub-quadratic speedup

Can we get a quadratic speedup ($\sim 1/\epsilon$) in some MLMC applications?

MLMC applied to:

Nested Expectation

Nested expectation

$$E_U\left[f\left(U,\, E_{V|U}[g(U,V)]\right)\right]$$

Nested expectation

$$E_U\left[f\left(U,\ E_{V|U}[g(U,V)]\right)\right]$$

Examples:

- Medical decision making:

$$E_U\left[\max_k E_{V|U}[g_k(U,V)]\right] - \max_k E[g_k(U,V)]$$

« Benefit of treatment k when making observation U »

- Pricing a compound option:

$$E_U\left[\max\left(E_{V|U}[\max(V-k,0)] - k,0\right)\right]$$

« Call on Call option »

Nested expectation

$$E_U\Big[f\Big(U,\ E_{V|U}[g(U,V)]\Big)\Big]$$

Examples:

- Medical decision making:

$$E_U\Big[\max_k E_{V|U}[g_k(U,V)]\Big] - \max_k E[g_k(U,V)]$$

« Benefit of treatment k when making observation U »

- Pricing a compound option:

$$E_U\Big[\max\Big(E_{V|U}[\max(V-k,0)] - k,0\Big)\Big]$$

« Call on Call option »

Assumptions

- Lipschitz:
  $$|f(u,x_1) - f(u,x_2)| \le |x_1 - x_2|$$
- Inner variance:
  $$\mathrm{Var}_{V|u}[g(u,V)] \le 1 \qquad (\forall u)$$
- Outer variance:
  $$\mathrm{Var}_U[f(U,\ E_{V|U}[g(U,V)])] \le 1$$

Nested expectation

$$E_U\left[f\left(U, \boxed{E_{V|U}[g(U,V)]}\right)\right]$$

MLMC: replace inner expectation with tunable stochastic process

Nested expectation

$$E_U\left[f\left(U, \boxed{E_{V|U}[g(U, V)]}\right)\right]$$

MLMC: replace inner expectation with tunable stochastic process

1. For $T = 1,2,4,8,\ldots,\lceil 1/\epsilon \rceil$:

   A. Generate $U_T \sim U$

   B. Estimate $X_T^1 \leftarrow E_{V|U_T}[g(U_T, V)]$ with error $1/T$

Nested expectation

$$E_U\left[f\left(U, \boxed{E_{V|U}[g(U,V)]}\right)\right]$$

MLMC: replace inner expectation with tunable stochastic process

1. For $T = 1,2,4,8,\ldots,\lceil 1/\epsilon \rceil$:

   A. Generate $U_T \sim U$

   B. Estimate $X_T^1 \leftarrow E_{V|U_T}[g(U_T, V)]$ with error $1/T$

   C. Estimate $X_T^2 \leftarrow E_{V|U_T}[g(U_T, V)]$ with error $1/(2T)$

   D. Return $Y_T = f(U_T, X_T^1) - f(U_T, X_T^2)$

Nested expectation

$$E_U\left[f\left(U, \boxed{E_{V|U}[g(U,V)]}\right)\right]$$

MLMC: replace inner expectation with tunable stochastic process

1. For $T = 1,2,4,8,\ldots,\lceil 1/\epsilon \rceil$:

   Estimate $\Delta_T \leftarrow E[Y_T]$ with error $\epsilon/\log(1/\epsilon)$ where Y_T is:

   A. Generate $U_T \sim U$
   B. Estimate $X_T^1 \leftarrow E_{V|U_T}[g(U_T, V)]$ with error $1/T$
   C. Estimate $X_T^2 \leftarrow E_{V|U_T}[g(U_T, V)]$ with error $1/(2T)$
   D. Return $Y_T = f(U_T, X_T^1) - f(U_T, X_T^2)$

Nested expectation

$$E_U\left[f\left(U, \boxed{E_{V|U}[g(U,V)]}\right)\right]$$

MLMC: replace inner expectation with tunable stochastic process

1. For $T = 1,2,4,8,\ldots, \lceil 1/\epsilon \rceil$:

   Estimate $\Delta_T \leftarrow E[Y_T]$ with error $\epsilon/\log(1/\epsilon)$ where Y_T is:

   A. Generate $U_T \sim U$
   B. Estimate $X_T^1 \leftarrow E_{V|U_T}[g(U_T, V)]$ with error $1/T$
   C. Estimate $X_T^2 \leftarrow E_{V|U_T}[g(U_T, V)]$ with error $1/(2T)$
   D. Return $Y_T = f(U_T, X_T^1) - f(U_T, X_T^2)$

2. Output $\mu = X_1^1 + \Delta_1 + \Delta_2 + \ldots + \Delta_{\lceil 1/\epsilon \rceil}$

Nested expectation

$$E_U\left[f\left(U, \boxed{E_{V|U}[g(U,V)]}\right)\right]$$

MLMC: replace inner expectation with tunable stochastic process

Classical est.

1. For $T = 1,2,4,8,\ldots,\lceil 1/\epsilon \rceil$: $\leftarrow$ $\log(1/\epsilon)$

$\times$

Estimate $\Delta_T \leftarrow E[Y_T]$ with error $\epsilon/\log(1/\epsilon)$ where Y_T is: $\leftarrow$ $\log^2(1/\epsilon)/(\epsilon T)^2$

$\times$

   A. Generate $U_T \sim U$
   B. Estimate $X_T^1 \leftarrow E_{V|U_T}[g(U_T, V)]$ with error $1/T$ $\leftarrow$ T^2

   $+$

   C. Estimate $X_T^2 \leftarrow E_{V|U_T}[g(U_T, V)]$ with error $1/(2T)$ $\leftarrow$ T^2
   D. Return $Y_T = f(U_T, X_T^1) - f(U_T, X_T^2)$

2. Output $\mu = X_1^1 + \Delta_1 + \Delta_2 + \ldots + \Delta_{\lceil 1/\epsilon \rceil}$

Nested expectation

$$E_U\left[f\left(U, \boxed{E_{V|U}[g(U,V)]}\right)\right]$$

MLMC: replace inner expectation with tunable stochastic process

Classical est.

1. For $T = 1,2,4,8,\ldots,\lceil 1/\epsilon \rceil$: $\longleftarrow$ $\log(1/\epsilon)$

$\times$

Estimate $\Delta_T \leftarrow E[Y_T]$ with error $\epsilon/\log(1/\epsilon)$ where Y_T is: $\longleftarrow$ $\log^2(1/\epsilon)/(\epsilon T)^2$

$\times$

A. Generate $U_T \sim U$

B. Estimate $X_T^1 \leftarrow E_{V|U_T}[g(U_T, V)]$ with error $1/T$ $\longleftarrow$ T^2

$+$

C. Estimate $X_T^2 \leftarrow E_{V|U_T}[g(U_T, V)]$ with error $1/(2T)$ $\longleftarrow$ T^2

D. Return $Y_T = f(U_T, X_T^1) - f(U_T, X_T^2)$

$\sim 1/\epsilon^2$

2. Output $\mu = X_1^1 + \Delta_1 + \Delta_2 + \ldots + \Delta_{\lceil 1/\epsilon \rceil}$

Nested expectation

$$E_U\left[f\left(U, \boxed{E_{V|U}[g(U,V)]}\right)\right]$$

MLMC: replace inner expectation with tunable stochastic process

		Classical est.	Quantum est.
1. For $T = 1,2,4,8,\ldots,\lceil 1/\epsilon \rceil$:	$\longleftarrow$	$\log(1/\epsilon)$	$\log(1/\epsilon)$
		$\times$	$\times$
Estimate $\Delta_T \leftarrow E[Y_T]$ with error $\epsilon/\log(1/\epsilon)$ where Y_T is:	$\longleftarrow$	$\log^2(1/\epsilon)/(\epsilon T)^2$	$\log(1/\epsilon)/(\epsilon T)$
		$\times$	$\times$
A. Generate $U_T \sim U$			
B. Estimate $X_T^1 \leftarrow E_{V\|U_T}[g(U_T,V)]$ with error $1/T$	$\longleftarrow$	T^2	T
		$+$	$+$
C. Estimate $X_T^2 \leftarrow E_{V\|U_T}[g(U_T,V)]$ with error $1/(2T)$	$\longleftarrow$	T^2	T
D. Return $Y_T = f(U_T, X_T^1) - f(U_T, X_T^2)$			
		$\sim 1/\epsilon^2$	$\sim 1/\epsilon$

2. Output $\mu = X_1^1 + \Delta_1 + \Delta_2 + \ldots + \Delta_{\lceil 1/\epsilon \rceil}$

Nested expectation

$$E_U\left[f\left(U,\, E_{V|U}[g(U,V)]\right)\right]$$

What permitted a full quadratic speedup?

The possibility of accelerating quadratically the generating stochastic process X_T itself.

Nested expectation

$$E_U\left[f\left(U,\, E_{V|U}[g(U,V)]\right)\right]$$

What permitted a full quadratic speedup?

The possibility of accelerating quadratically the generating stochastic process X_T itself.

Obstacle to a similar speedup for SDE:

We don't know quantum algorithms for computing the T-th iterate X_T of a discretization scheme (e.g., Euler method) faster than in time T.

Are there interesting SDE for which this can be accelerated?