Out-of-distribution (OOD) generalization for learning quantum dynamics and dynamical simulation

Matthias C. Caro

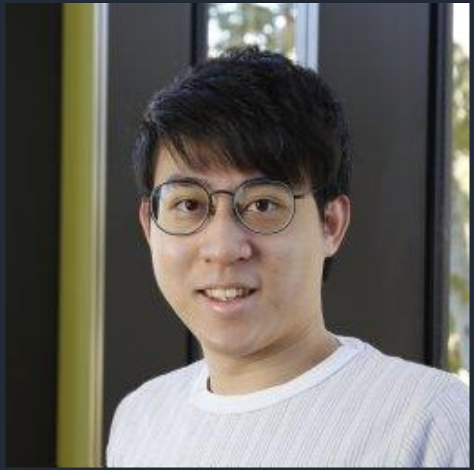
Caltech

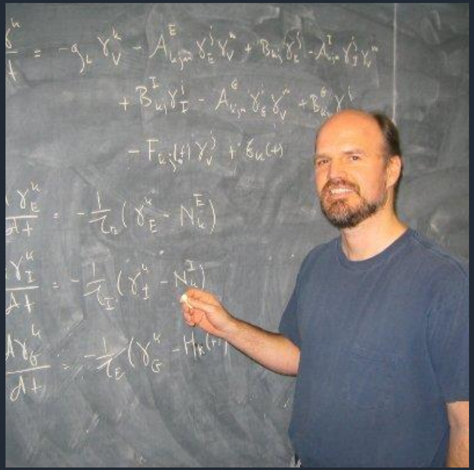
Freie Universität Berlin

TQC 2023

Based on arXiv:2204.10268 and arXiv:2204.10269

My collaborators





Motivation

What this talk is about

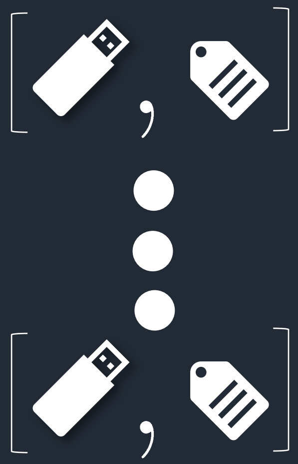
In-Distribution Generalization – Intuition

In-Distribution Generalization – Intuition

Phase 1: Training

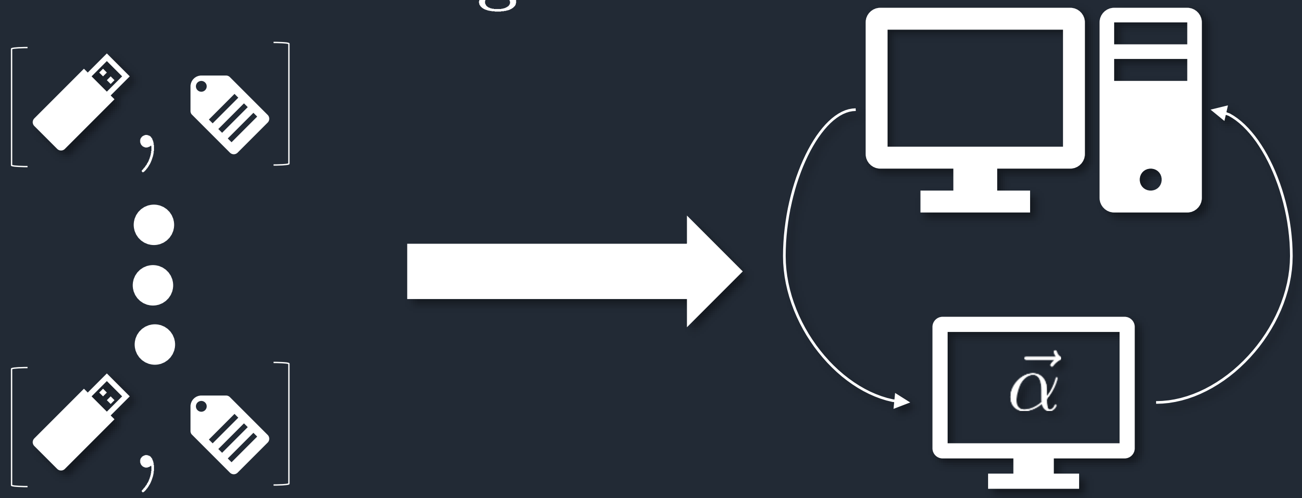
In-Distribution Generalization – Intuition

Phase 1: Training



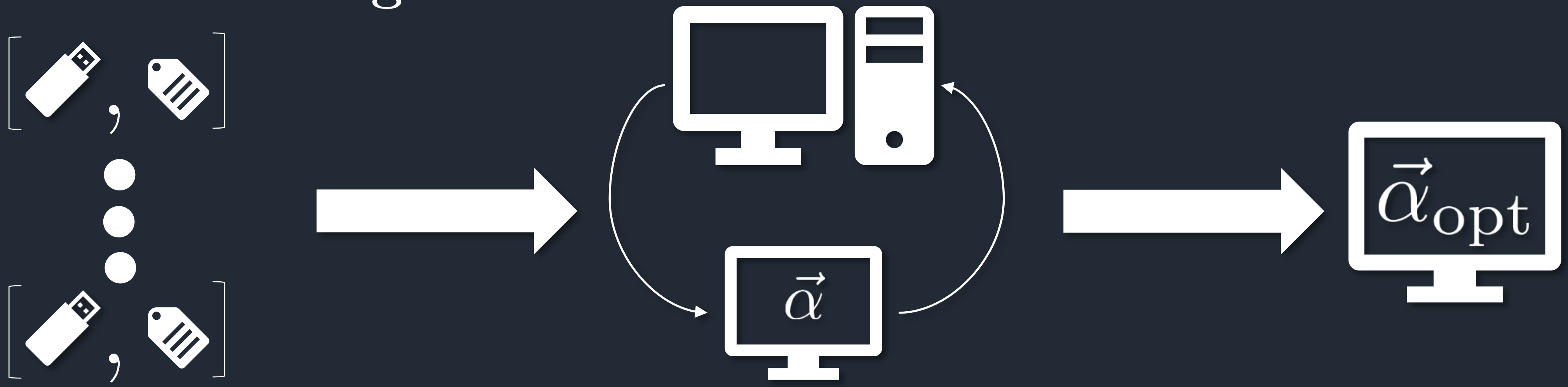
In-Distribution Generalization – Intuition

Phase 1: Training



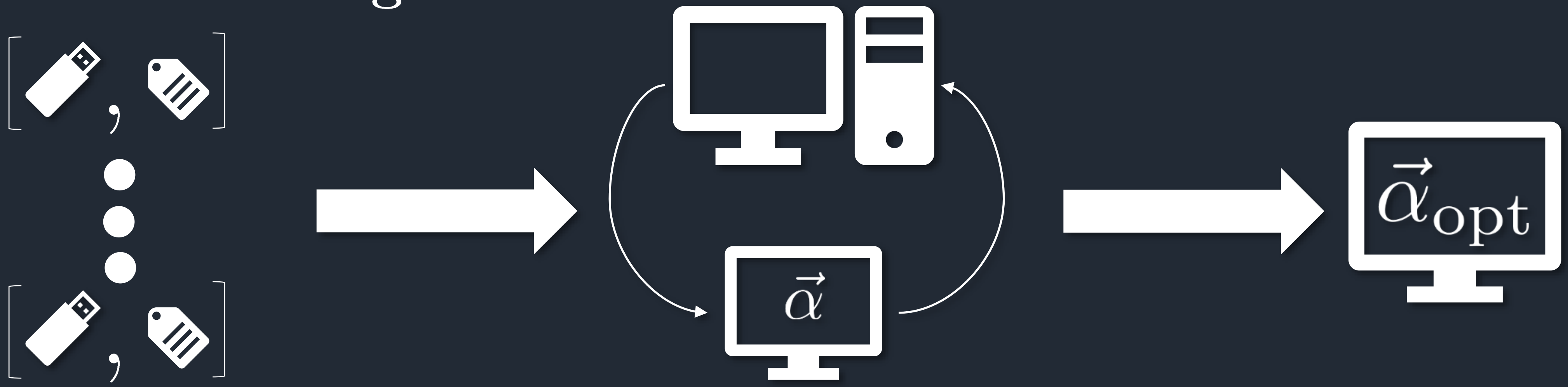
In-Distribution Generalization – Intuition

Phase 1: Training



In-Distribution Generalization – Intuition

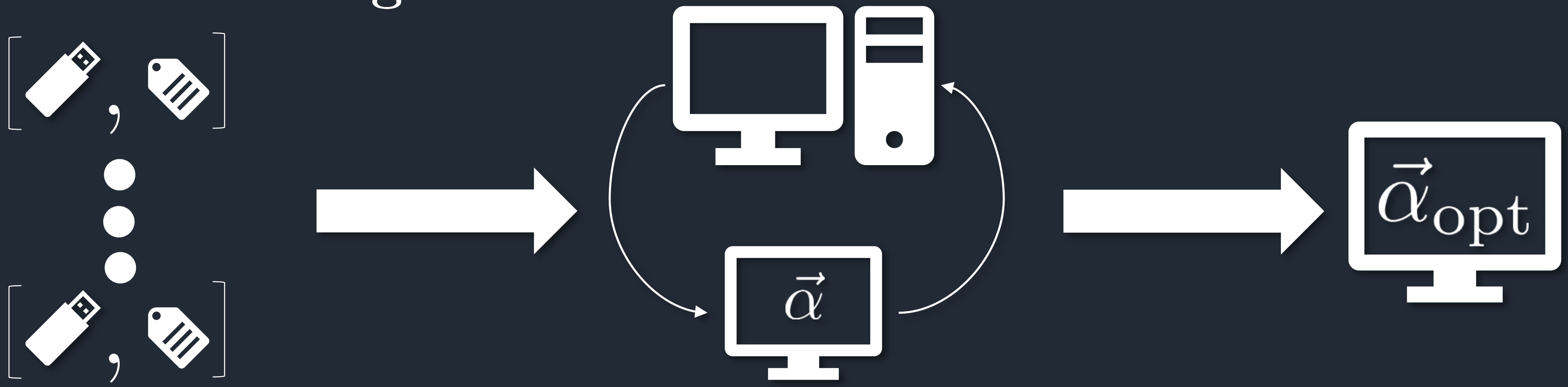
Phase 1: Training



Phase 2: Testing on data from the same source

In-Distribution Generalization – Intuition

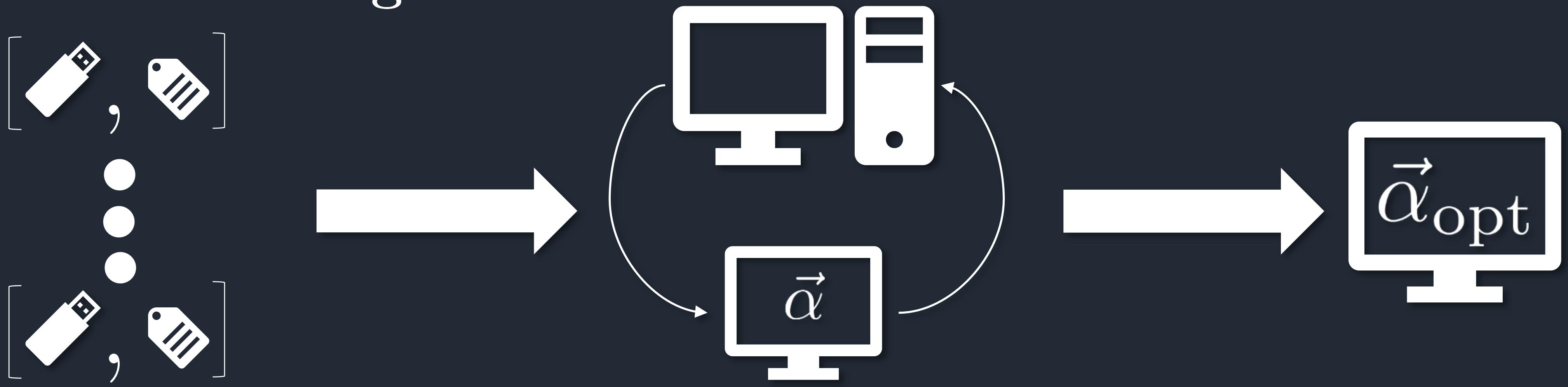
Phase 1: Training



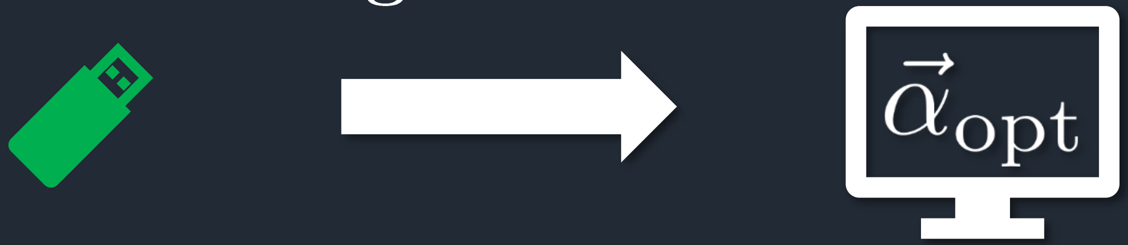
Phase 2: Testing on data from the same source

In-Distribution Generalization – Intuition

Phase 1: Training

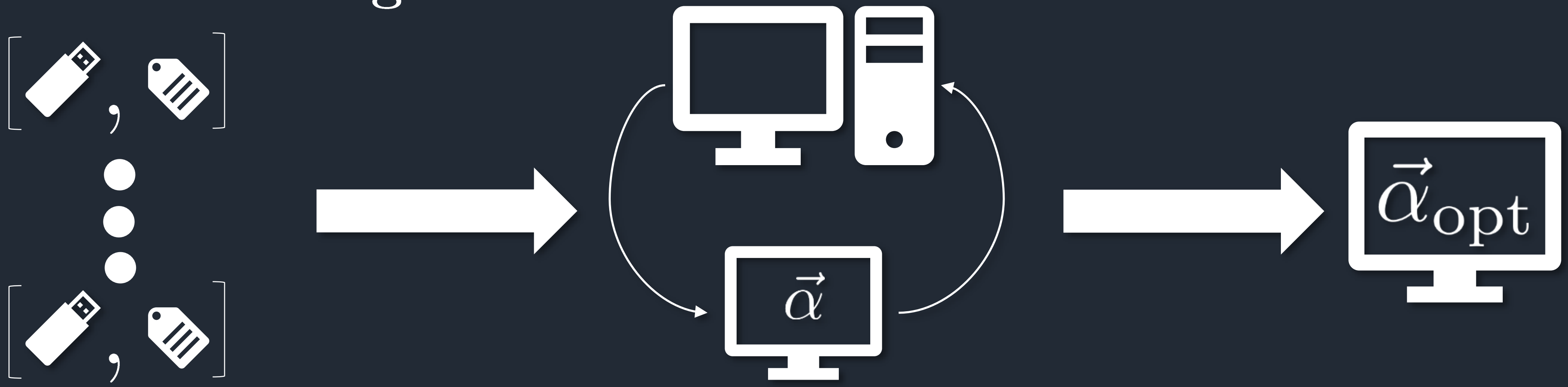


Phase 2: Testing on data from the same source



In-Distribution Generalization – Intuition

Phase 1: Training

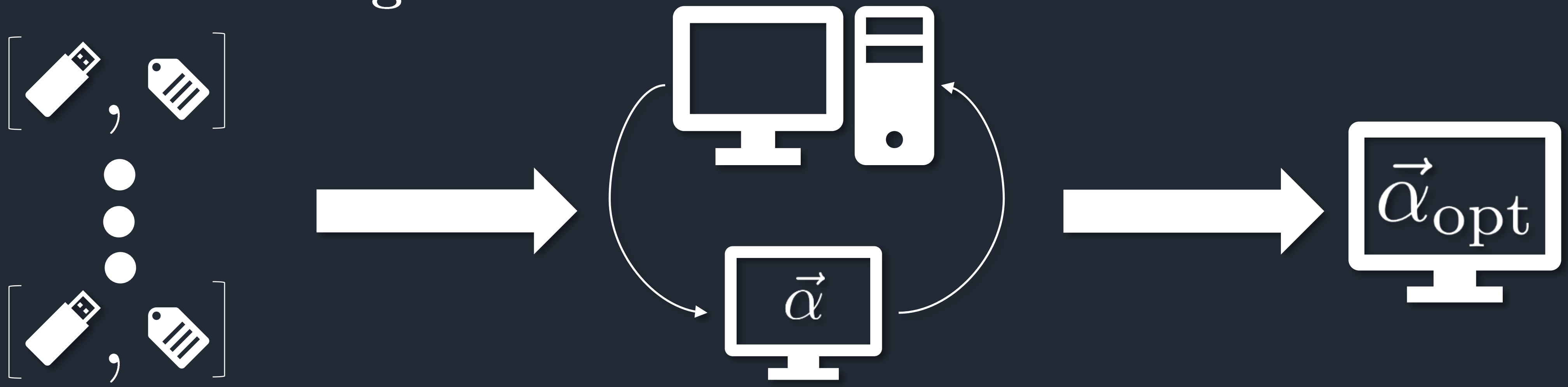


Phase 2: Testing on data from the same source

OOD Generalization – Intuition

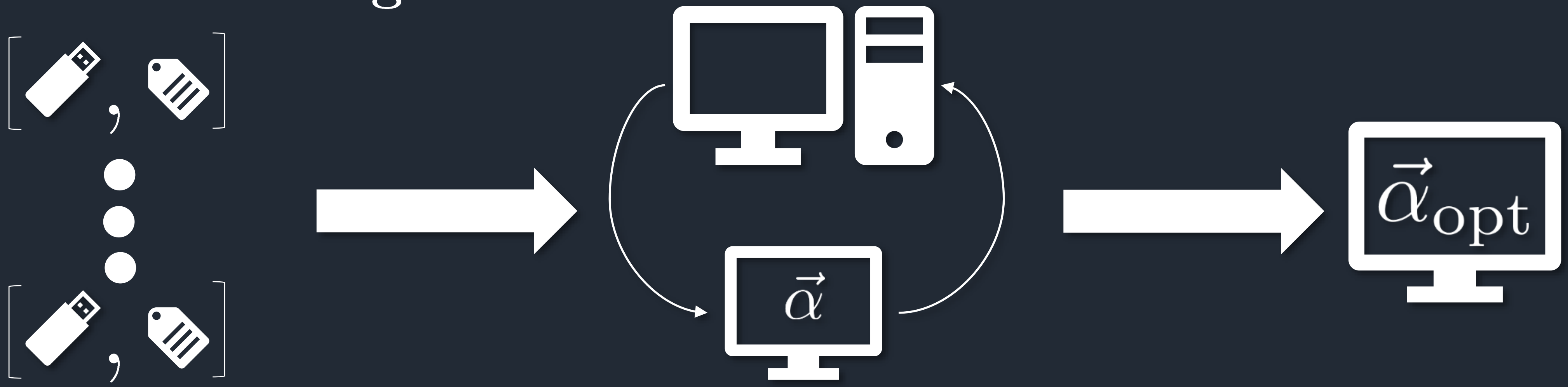
OOD Generalization – Intuition

Phase 1: Training



OOD Generalization – Intuition

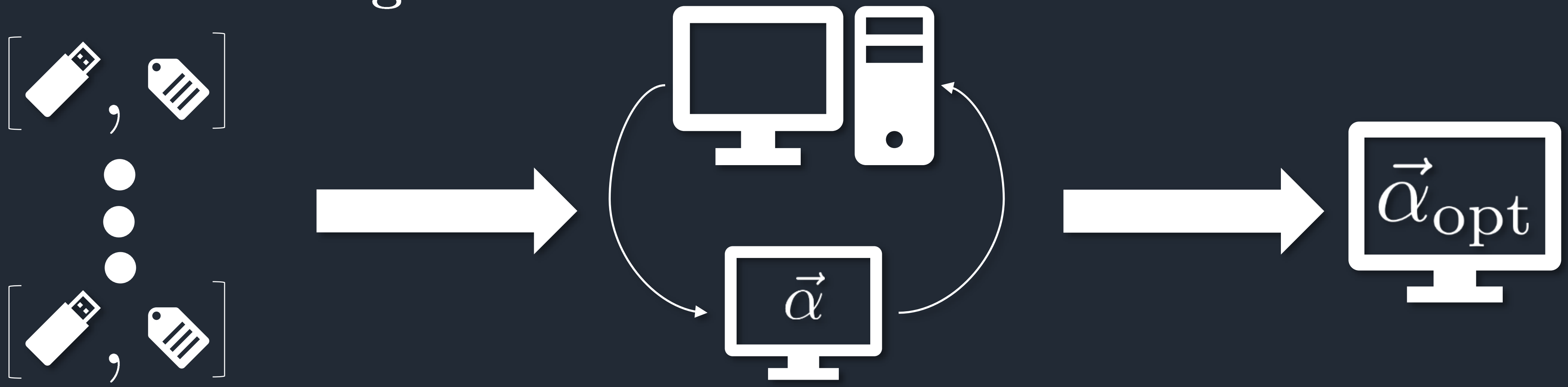
Phase 1: Training



Phase 2: Testing on data from a different source

OOD Generalization – Intuition

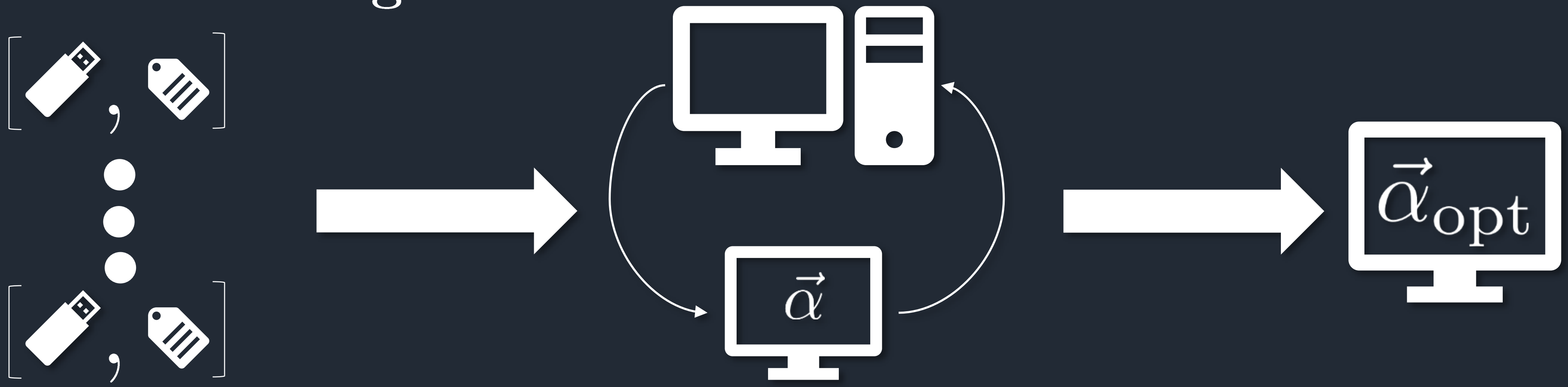
Phase 1: Training



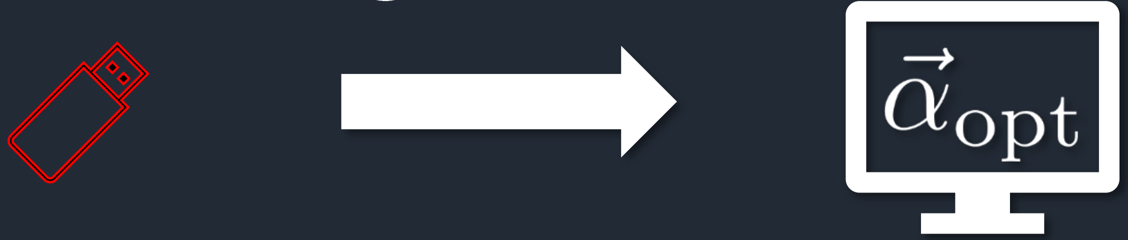
Phase 2: Testing on data from a different source

OOD Generalization – Intuition

Phase 1: Training

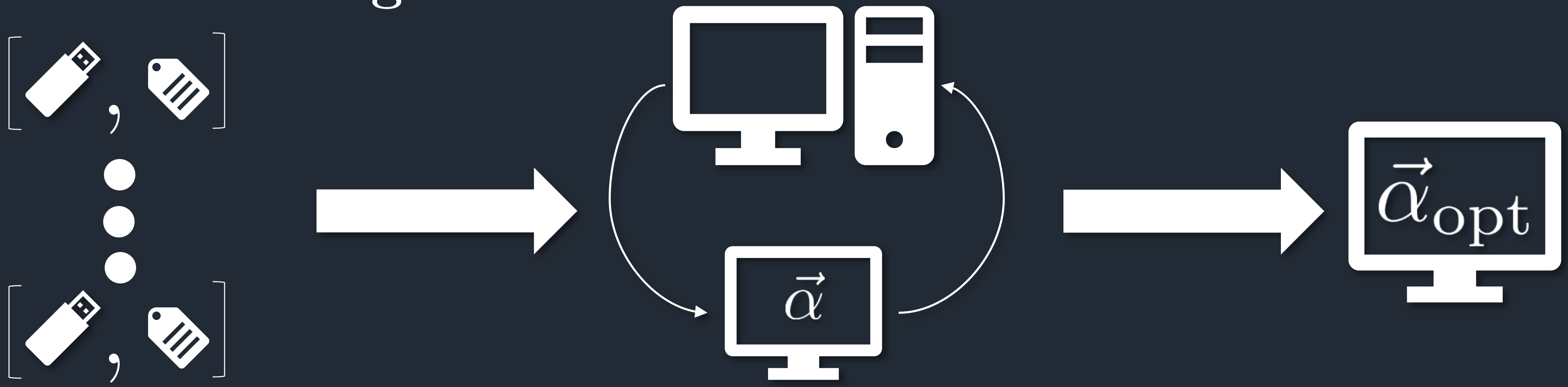


Phase 2: Testing on data from a different source



OOD Generalization – Intuition

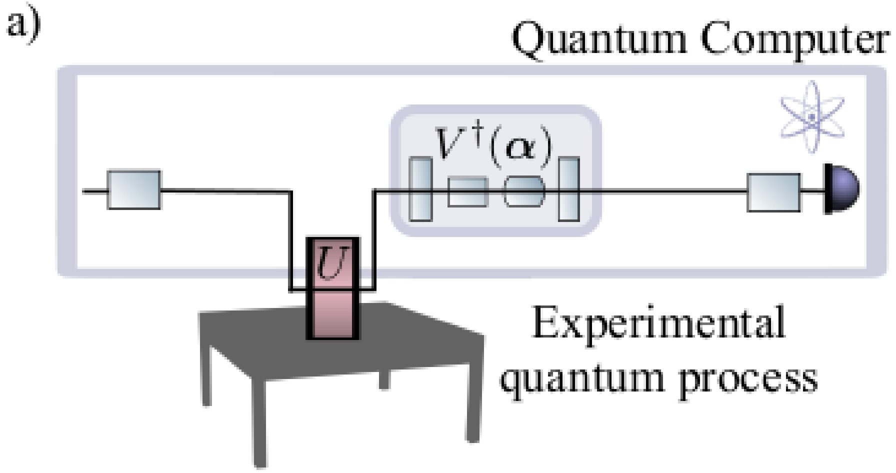
Phase 1: Training



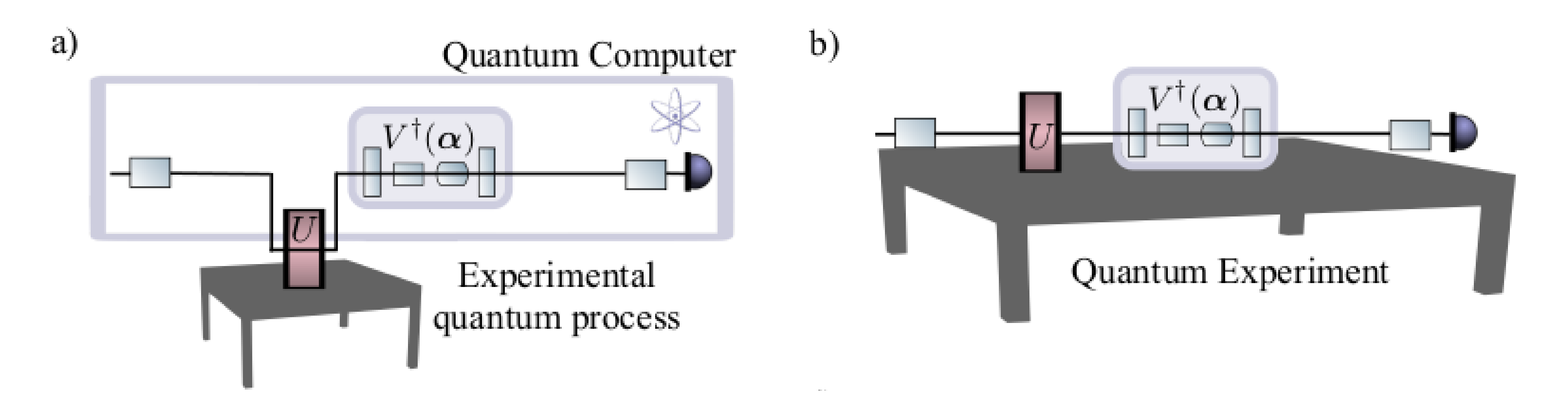
Phase 2: Testing on data from a different source

Use Cases for Unitary Learning

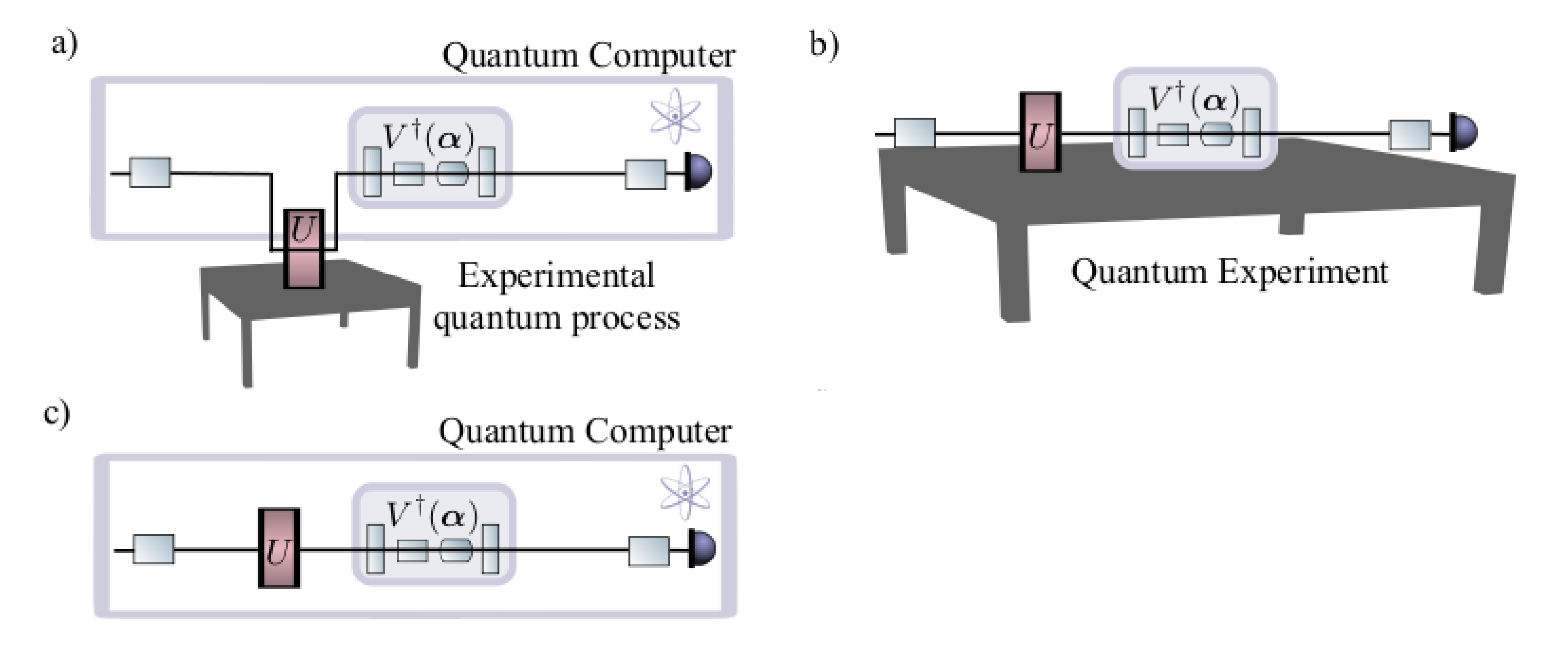
Use Cases for Unitary Learning



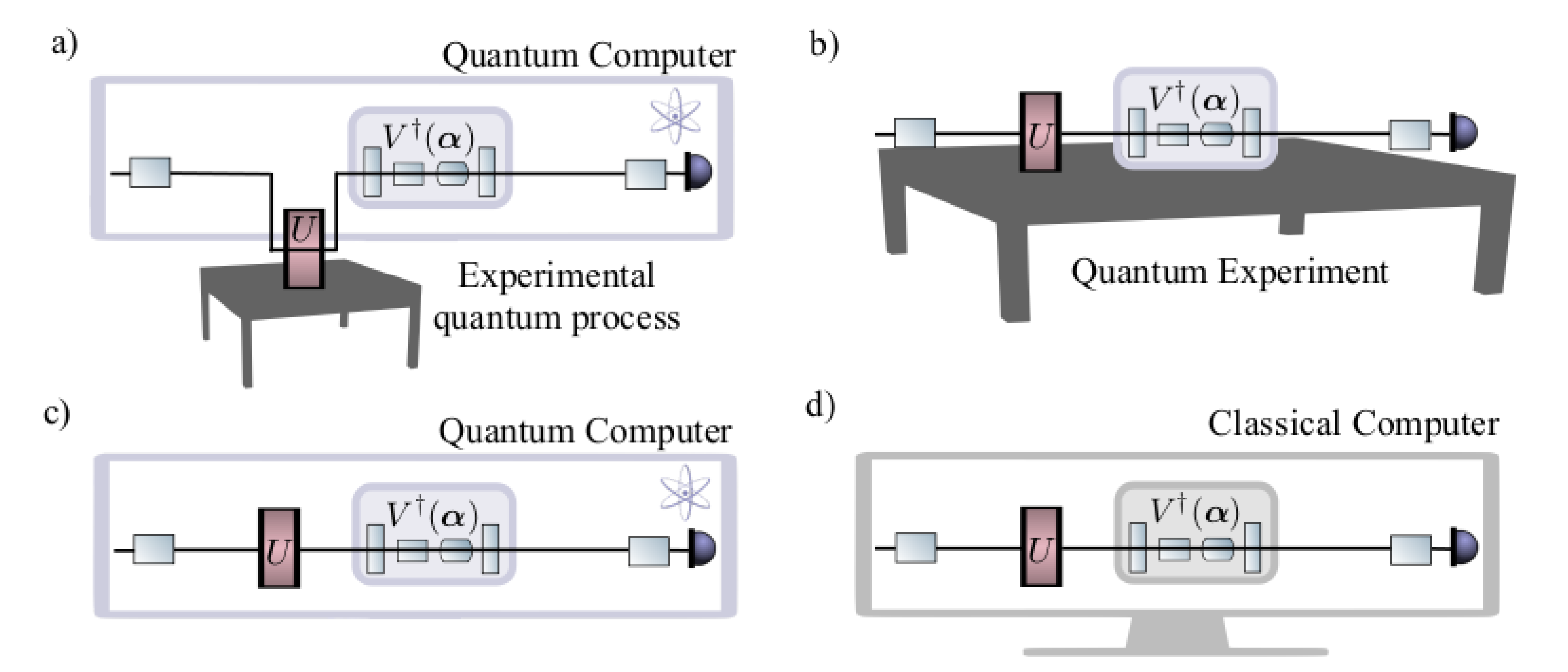
Use Cases for Unitary Learning



Use Cases for Unitary Learning



Use Cases for Unitary Learning



Outline

Framework and Problem Setup

What learning problem we consider

Framework – High Level View

Framework – High Level View

Goal: Learn an unknown $\mathsf{U} \in \mathcal{U}\left(\left(\mathbb{C}^2\right)^{\otimes n}\right)$.

Framework – High Level View

Goal: Learn an unknown $\mathsf{U} \in \mathcal{U}\left(\left(\mathbb{C}^2\right)^{\otimes n}\right)$.

Ansatz: Unitary QNN $V(\vec{\alpha})$ with trainable classical param. $\vec{\alpha}$

Framework – High Level View

Goal: Learn an unknown $\mathsf{U} \in \mathcal{U}\left(\left(\mathbb{C}^2\right)^{\otimes n}\right)$.

Ansatz: Unitary QNN $V(\vec{\alpha})$ with trainable classical param. $\vec{\alpha}$

Strategy: Use quantum computer to evaluate whether a param. setting $\vec{\alpha}$ is good on available data and in which direction you can improve.

Framework – High Level View

Goal: Learn an unknown $\mathrm{U} \in \mathcal{U}\left(\left(\mathbb{C}^2\right)^{\otimes n}\right)$.

Ansatz: Unitary QNN $V(\vec{\alpha})$ with trainable classical param. $\vec{\alpha}$

Strategy: Use quantum computer to evaluate whether a param. setting $\vec{\alpha}$ is good on available data and in which direction you can improve.

Use classical computer for the actual optimization of the parameters, obtaining $\vec{\alpha}_{opt}$.

Mathematical Problem Formulation

Mathematical Problem Formulation

Definition (Expected Testing Risk):

Mathematical Problem Formulation

Definition (Expected Testing Risk):

If U is the unknown unitary to be learned, the *expected testing risk* of the parameter setting $\vec{\alpha}$ of a QNN $V(\vec{\alpha})$ w.r.t. the *testing ensemble* $\mathcal{P}$ is:

Mathematical Problem Formulation

Definition (Expected Testing Risk):

If U is the unknown unitary to be learned, the *expected testing risk* of the parameter setting $\vec{\alpha}$ of a QNN $V(\vec{\alpha})$ w.r.t. the *testing ensemble* $\mathcal{P}$ is:

$$R_{\mathcal{P}}(\vec{\alpha}) = \frac{1}{4}\mathbb{E}_{|\Psi\rangle\sim\mathcal{P}}\left[\left\|U|\Psi\rangle\langle\Psi|U^\dagger - V(\vec{\alpha})|\Psi\rangle\langle\Psi|V(\vec{\alpha})^\dagger\right\|_1^2\right].$$

Mathematical Problem Formulation

Definition (Expected Testing Risk):

If U is the unknown unitary to be learned, the *expected testing risk* of the parameter setting $\vec{\alpha}$ of a QNN $V(\vec{\alpha})$ w.r.t. the *testing ensemble* $\mathcal{P}$ is:

$$R_{\mathcal{P}}(\vec{\alpha}) = \frac{1}{4}\mathbb{E}_{|\Psi\rangle\sim\mathcal{P}}\left[\left\| U\,|\Psi\rangle\langle\Psi|\,U^\dagger - V(\vec{\alpha})\,|\Psi\rangle\langle\Psi|\,V(\vec{\alpha})^\dagger \right\|_1^2\right].$$

Goal: small $R_{\mathcal{P}}(\vec{\alpha}_{opt})$

Mathematical Problem Formulation

Definition (Expected Testing Risk):

If U is the unknown unitary to be learned, the *expected testing risk* of the parameter setting $\vec{\alpha}$ of a QNN $V(\vec{\alpha})$ w.r.t. the *testing ensemble* $\mathcal{P}$ is:

$$R_{\mathcal{P}}(\vec{\alpha}) = \frac{1}{4}\mathbb{E}_{|\Psi\rangle\sim\mathcal{P}}\left[\left\| U|\Psi\rangle\langle\Psi|U^\dagger - V(\vec{\alpha})|\Psi\rangle\langle\Psi|V(\vec{\alpha})^\dagger\right\|_1^2\right].$$

Goal: small $R_{\mathcal{P}}(\vec{\alpha}_{opt})$

But: As learner, we know neither U nor $\mathcal{P}$...

Mathematical Problem Formulation

Mathematical Problem Formulation

Definition (Training Cost):

Mathematical Problem Formulation

Definition (Training Cost):

Given training data $\mathcal{D}_{\mathcal{Q}}(N) = \{|\Psi^{(j)}\rangle, U|\Psi^{(j)}\rangle\}_{j=1}^{N}$, where each $|\Psi^{(j)}\rangle$ is drawn i.i.d. from the *training ensemble* $\mathcal{Q}$, the *training cost* of the parameter setting $\vec{\alpha}$ of a QNN $V(\vec{\alpha})$ is:

Mathematical Problem Formulation

Definition (Training Cost):

Given training data $\mathcal{D}_{\mathcal{Q}}(N) = \{|\Psi^{(j)}\rangle, U|\Psi^{(j)}\rangle\}_{j=1}^{N}$, where each $|\Psi^{(j)}\rangle$ is drawn i.i.d. from the *training ensemble* $\mathcal{Q}$, the *training cost* of the parameter setting $\vec{\alpha}$ of a QNN $V(\vec{\alpha})$ is:

$$C_{\mathcal{D}_{\mathcal{Q}}(N)}(\vec{\alpha}) = \frac{1}{4N} \sum_{j=1}^{N} \left\| U \left|\Psi^{(j)}\right\rangle \left\langle\Psi^{(j)}\right| U^\dagger - V(\vec{\alpha}) \left|\Psi^{(j)}\right\rangle \left\langle\Psi^{(j)}\right| V(\vec{\alpha})^\dagger \right\|_1^2 .$$

Mathematical Problem Formulation

Definition (Training Cost):

Given training data $\mathcal{D}_{\mathcal{Q}}(N) = \{|\Psi^{(j)}\rangle, U|\Psi^{(j)}\rangle\}_{j=1}^{N}$, where each $|\Psi^{(j)}\rangle$ is drawn i.i.d. from the *training ensemble* $\mathcal{Q}$, the *training cost* of the parameter setting $\vec{\alpha}$ of a QNN $V(\vec{\alpha})$ is:

$$C_{\mathcal{D}_{\mathcal{Q}}(N)}(\vec{\alpha}) = \frac{1}{4N} \sum_{j=1}^{N} \left\| U \left|\Psi^{(j)}\right\rangle \left\langle\Psi^{(j)}\right| U^{\dagger} - V(\vec{\alpha}) \left|\Psi^{(j)}\right\rangle \left\langle\Psi^{(j)}\right| V(\vec{\alpha})^{\dagger} \right\|_{1}^{2}.$$

Idea: Small $C_{\mathcal{D}_{\mathcal{Q}}(N)}(\vec{\alpha})$ as proxy for small $R_{\mathcal{P}}(\vec{\alpha})$

Mathematical Problem Formulation

Definition (Training Cost):

Given training data $\mathcal{D}_{\mathcal{Q}}(N) = \{|\Psi^{(j)}\rangle, U|\Psi^{(j)}\rangle\}_{j=1}^N$, where each $|\Psi^{(j)}\rangle$ is drawn i.i.d. from the *training ensemble* $\mathcal{Q}$, the *training cost* of the parameter setting $\vec{\alpha}$ of a QNN $V(\vec{\alpha})$ is:

$$C_{\mathcal{D}_{\mathcal{Q}}(N)}(\vec{\alpha}) = \frac{1}{4N}\sum_{j=1}^N \left\| U|\Psi^{(j)}\rangle\langle\Psi^{(j)}|U^\dagger - V(\vec{\alpha})|\Psi^{(j)}\rangle\langle\Psi^{(j)}|V(\vec{\alpha})^\dagger \right\|_1^2 .$$

Idea: Small $C_{\mathcal{D}_{\mathcal{Q}}(N)}(\vec{\alpha})$ as proxy for small $R_{\mathcal{P}}(\vec{\alpha})$

But: When is it indeed a good proxy?

Mathematical Problem Formulation

Idea: Small $C_{\mathcal{D}_{\mathcal{Q}}(N)}(\vec{\alpha})$ as proxy for small $R_{\mathcal{P}}(\vec{\alpha})$

But: When is it indeed a good proxy?

Mathematical Problem Formulation

Idea: Small $C_{\mathcal{D}_{\mathcal{Q}}(N)}(\vec{\alpha})$ as proxy for small $R_{\mathcal{P}}(\vec{\alpha})$

But: When is it indeed a good proxy?

Question of Generalization:

Mathematical Problem Formulation

Idea: Small $C_{\mathcal{D}_{\mathcal{Q}}(N)}(\vec{\alpha})$ as proxy for small $R_{\mathcal{P}}(\vec{\alpha})$

But: When is it indeed a good proxy?

Question of Generalization:

When can we guarantee "small $C_{\mathcal{D}_{\mathcal{Q}}(N)}\left(\vec{\alpha}_{opt}\right) \Rightarrow$ small $R_{\mathcal{P}}\left(\vec{\alpha}_{opt}\right)$"?

Mathematical Problem Formulation

Idea: Small $C_{\mathcal{D}_{\mathcal{Q}}(N)}(\vec{\alpha})$ as proxy for small $R_{\mathcal{P}}(\vec{\alpha})$

But: When is it indeed a good proxy?

Question of Generalization:

When can we guarantee "small $C_{\mathcal{D}_{\mathcal{Q}}(N)}(\vec{\alpha}_{opt}) \Rightarrow$ small $R_{\mathcal{P}}(\vec{\alpha}_{opt})$"?

Two variants of the question:

Mathematical Problem Formulation

Idea: Small $C_{\mathcal{D}_{\mathcal{Q}}(N)}(\vec{\alpha})$ as proxy for small $R_{\mathcal{P}}(\vec{\alpha})$

But: When is it indeed a good proxy?

Question of Generalization:

When can we guarantee "small $C_{\mathcal{D}_{\mathcal{Q}}(N)}(\vec{\alpha}_{opt}) \Rightarrow$ small $R_{\mathcal{P}}(\vec{\alpha}_{opt})$"?

Two variants of the question:

- $\mathcal{P} = \mathcal{Q}$: In-distribution (ID) generalization

Mathematical Problem Formulation

Idea: Small $C_{\mathcal{D}_{\mathcal{Q}}(N)}(\vec{\alpha})$ as proxy for small $R_{\mathcal{P}}(\vec{\alpha})$

But: When is it indeed a good proxy?

Question of Generalization:

When can we guarantee "small $C_{\mathcal{D}_{\mathcal{Q}}(N)}(\vec{\alpha}_{opt}) \Rightarrow$ small $R_{\mathcal{P}}(\vec{\alpha}_{opt})$"?

Two variants of the question:

- $\mathcal{P} = \mathcal{Q}$: In-distribution (ID) generalization
- $\mathcal{P} \neq \mathcal{Q}$: Out-of-distribution (OOD) generalization

Locally Scrambled Ensembles

What data sources we consider

Locally Scrambled Ensembles

Locally Scrambled Ensembles

Question: What should $\mathcal{Q}$ and $\mathcal{P}$ have in common for OOD generalization from $\mathcal{Q}$ to $\mathcal{P}$ to be possible?

Locally Scrambled Ensembles

Question: What should $\mathcal{Q}$ and $\mathcal{P}$ have in common for OOD generalization from $\mathcal{Q}$ to $\mathcal{P}$ to be possible?

Definition (Locally Scrambled Ensembles [1, 2]):

[1] W.-T. Kuo, A.A. Akhtar, D.P. Arovas, and Y.-Z. You; *Phys. Rev. B 101, 224202 (2020)*
[2] H.-Y. Hu, S. Choi, and Y.-Z. You; *Phys. Rev. Research 5, 023027 (2023)*

Locally Scrambled Ensembles

Question: What should $\mathcal{Q}$ and $\mathcal{P}$ have in common for OOD generalization from $\mathcal{Q}$ to $\mathcal{P}$ to be possible?

Definition (Locally Scrambled Ensembles [1, 2]):

- $\mathcal{U}_{LS} \in \mathbb{U}_{LS} :\Leftrightarrow \mathcal{U}_{LS}(\bigotimes_{i=1}^{n} U_i) = \mathcal{U}_{LS}$ for all $U_i \in \mathcal{U}(\mathbb{C}^2)$.

[1] W.-T. Kuo, A.A. Akhtar, D.P. Arovas, and Y.-Z. You; *Phys. Rev. B 101, 224202 (2020)*
[2] H.-Y. Hu, S. Choi, and Y.-Z. You; *Phys. Rev. Research 5, 023027 (2023)*

Locally Scrambled Ensembles

Question: What should $\mathcal{Q}$ and $\mathcal{P}$ have in common for OOD generalization from $\mathcal{Q}$ to $\mathcal{P}$ to be possible?

Definition (Locally Scrambled Ensembles [1, 2]):

- $\mathcal{U}_{LS} \in \mathbb{U}_{LS} :\Leftrightarrow \mathcal{U}_{LS}(\bigotimes_{i=1}^{n} U_i) = \mathcal{U}_{LS}$ for all $U_i \in \mathcal{U}(\mathbb{C}^2)$.
- $\mathcal{S}_{LS} \in \mathbb{S}_{LS} :\Leftrightarrow \mathcal{S}_{LS} = \mathcal{U}_{LS}\,|0\rangle^{\otimes n}$ for some $\mathcal{U}_{LS} \in \mathbb{U}_{LS}$.

[1] W.-T. Kuo, A.A. Akhtar, D.P. Arovas, and Y.-Z. You; *Phys. Rev. B 101, 224202 (2020)*
[2] H.-Y. Hu, S. Choi, and Y.-Z. You; *Phys. Rev. Research 5, 023027 (2023)*

Locally Scrambled Ensembles

Question: What should $\mathcal{Q}$ and $\mathcal{P}$ have in common for OOD generalization from $\mathcal{Q}$ to $\mathcal{P}$ to be possible?

Definition (Locally Scrambled Ensembles [1, 2]):

- $\mathcal{U}_{LS} \in \mathbb{U}_{LS} :\Leftrightarrow \mathcal{U}_{LS}(\bigotimes_{i=1}^{n} U_i) = \mathcal{U}_{LS}$ for all $U_i \in \mathcal{U}(\mathbb{C}^2)$.
- $\mathcal{S}_{LS} \in \mathbb{S}_{LS} :\Leftrightarrow \mathcal{S}_{LS} = \mathcal{U}_{LS}\,|0\rangle^{\otimes n}$ for some $\mathcal{U}_{LS} \in \mathbb{U}_{LS}$.

Intuition: Locally scrambled ensembles of unitaries/states

[1] W.-T. Kuo, A.A. Akhtar, D.P. Arovas, and Y.-Z. You; *Phys. Rev. B 101, 224202 (2020)*
[2] H.-Y. Hu, S. Choi, and Y.-Z. You; *Phys. Rev. Research 5, 023027 (2023)*

Locally Scrambled Ensembles

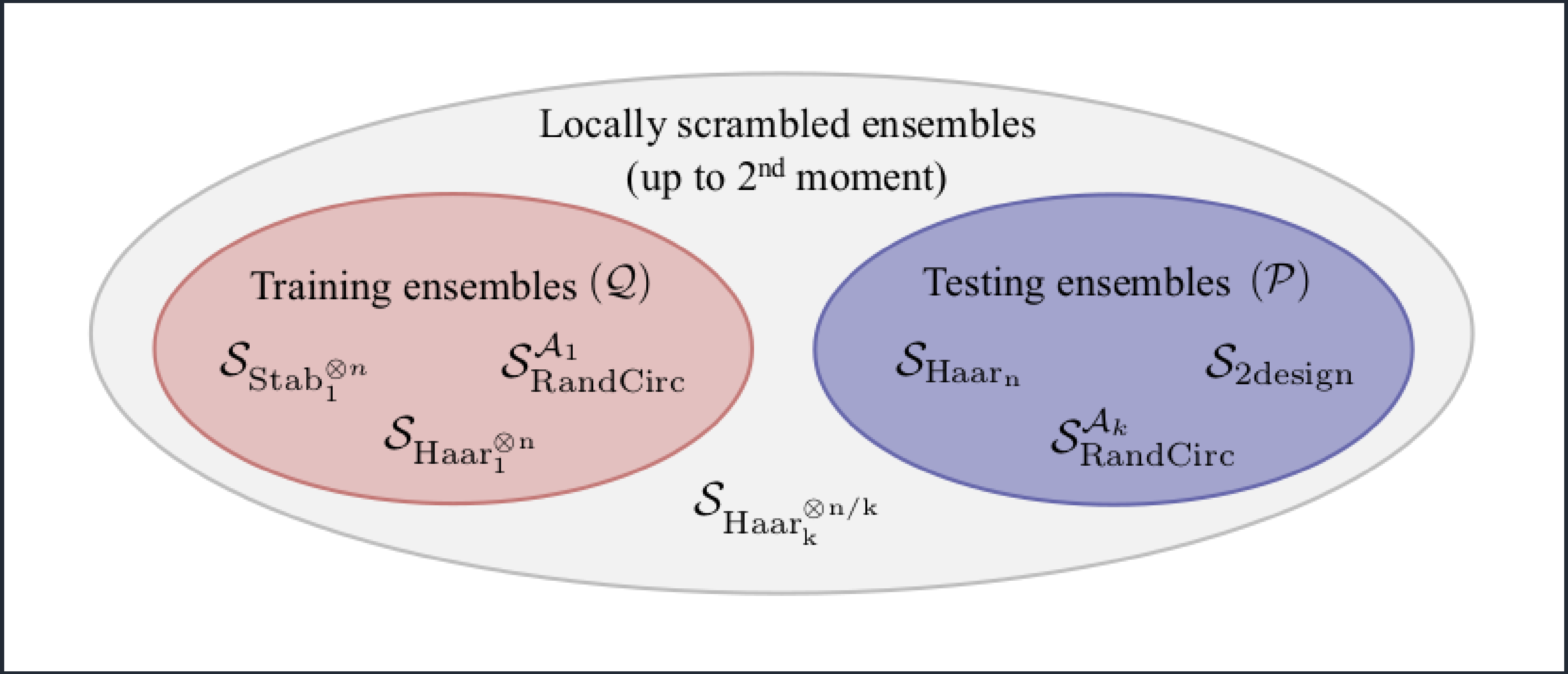
Question: What should $\mathcal{Q}$ and $\mathcal{P}$ have in common for OOD generalization from $\mathcal{Q}$ to $\mathcal{P}$ to be possible?

Definition (Locally Scrambled Ensembles [1, 2]):
- $\mathcal{U}_{LS} \in \mathbb{U}_{LS} :\Leftrightarrow \mathcal{U}_{LS}(\otimes_{i=1}^{n} U_i) = \mathcal{U}_{LS}$ for all $U_i \in \mathcal{U}(\mathbb{C}^2)$.
- $\mathcal{S}_{LS} \in \mathbb{S}_{LS} :\Leftrightarrow \mathcal{S}_{LS} = \mathcal{U}_{LS}\,|0\rangle^{\otimes n}$ for some $\mathcal{U}_{LS} \in \mathbb{U}_{LS}$.

Intuition: Locally scrambled ensembles of unitaries/states ≈ "at least locally uniformly random"

[1] W.-T. Kuo, A.A. Akhtar, D.P. Arovas, and Y.-Z. You; *Phys. Rev. B 101, 224202 (2020)*
[2] H.-Y. Hu, S. Choi, and Y.-Z. You; *Phys. Rev. Research 5, 023027 (2023)*

Locally Scrambled Ensembles – Training and Testing



Equivalence of Locally Scrambled Risks

How different locally scrambled risks are related

Equivalence of locally scrambled risks in unitary learning

Equivalence of locally scrambled risks in unitary learning

Theorem (Equivalence of locally scrambled ensembles for comparing unitaries [3]):

Let $\mathcal{P} \in \mathbb{S}_{\mathrm{LS}}^{(2)}$ and $\mathcal{Q} \in \mathbb{S}_{\mathrm{LS}}^{(2)}$, then for any parameter setting $\vec{\alpha}$,

$$\frac{1}{2} R_{\mathcal{Q}}\left(\vec{\alpha}\right) \leq R_{\mathcal{P}}\left(\vec{\alpha}\right) \leq 2 R_{\mathcal{Q}}\left(\vec{\alpha}\right) .$$

[3] **M.C.C.**, H.-Y. Huang, N. Ezzell, J. Gibbs, A.T. Sornborger, L. Cincio, P.J. Coles, Z. Holmes; *arXiv:2204.10268 (2022)*

Equivalence of locally scrambled risks in unitary learning

Theorem (Equivalence of locally scrambled ensembles for comparing unitaries [3]):

Let $\mathcal{P} \in \mathbb{S}_{\mathrm{LS}}^{(2)}$ and $\mathcal{Q} \in \mathbb{S}_{\mathrm{LS}}^{(2)}$, then for any parameter setting $\vec{\alpha}$,

$$\frac{1}{2} R_{\mathcal{Q}}\left(\vec{\alpha}\right) \leq R_{\mathcal{P}}\left(\vec{\alpha}\right) \leq 2 R_{\mathcal{Q}}\left(\vec{\alpha}\right) \, .$$

In words:
Any two locally scrambled risks differ by at most a constant factor.

[3] **M.C.C.**, H.-Y. Huang, N. Ezzell, J. Gibbs, A.T. Sornborger, L. Cincio, P.J. Coles, Z. Holmes; *arXiv:2204.10268 (2022)*

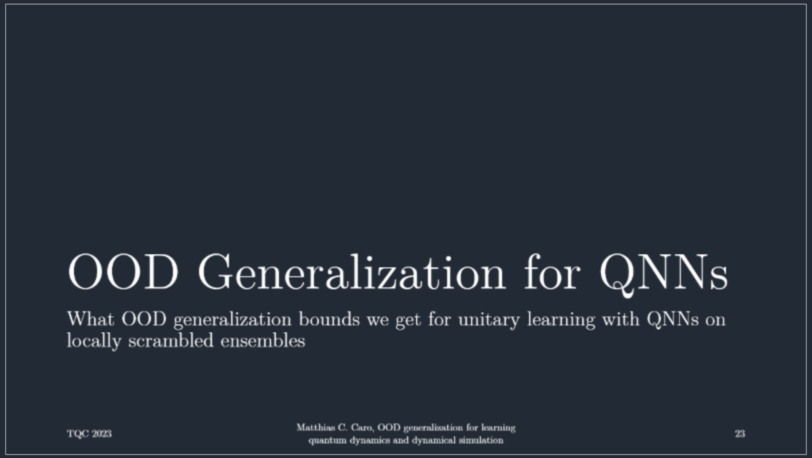
Proof Sketch

Proof Sketch

Do we have time for a proof sketch?

Proof Sketch

Do we have time for a proof sketch?



Proof Sketch:

Proof Sketch:

Our main technical result:

Proof Sketch:

Our main technical result:
Any locally scrambled risk is equivalent to the risk induced by the n-qubit Haar measure:

Proof Sketch:

Our main technical result:
Any locally scrambled risk is equivalent to the risk induced by the n-qubit Haar measure:

Lemma:
For any $\mathcal{Q} \in \mathbb{S}_{\mathrm{LS}}^{(2)}$ and any parameter setting $\vec{\alpha}$,

$$\frac{1}{2} R_{\mathcal{S}_{\mathrm{Haar}_n}}(\vec{\alpha}) \leq \frac{2^n}{2^n+1} R_{\mathcal{Q}}(\vec{\alpha}) \leq R_{\mathcal{S}_{\mathrm{Haar}_n}}(\vec{\alpha}) .$$

Proof Sketch:

Proof Sketch:

Proof of $\frac{1}{2} R_{\mathcal{S}_{\mathrm{Haar}_n}}(\vec{\alpha}) \leq \frac{2^n}{2^n+1} R_{\mathcal{Q}}(\vec{\alpha})$:

Proof Sketch:

Proof of $\frac{1}{2} R_{\mathcal{S}_{\mathrm{Haar}_n}}(\vec{\alpha}) \leq \frac{2^n}{2^n+1} R_{\mathcal{Q}}(\vec{\alpha})$**:**

- $\boldsymbol{n}$-qubit Haar risk vs squared Hilbert-Schmidt inner product:

Proof Sketch:

Proof of $\frac{1}{2}R_{\mathcal{S}_{\mathrm{Haar}_n}}(\vec{\alpha}) \leq \frac{2^n}{2^n+1}R_{\mathcal{Q}}(\vec{\alpha})$**:**

- n-qubit Haar risk vs squared Hilbert-Schmidt inner product:

$$R_{\mathcal{S}_{\mathrm{Haar}_n}}(\vec{\alpha}) = \frac{2^n}{2^n+1}\left(1 - \frac{1}{4^n}|\operatorname{tr}[U^\dagger V(\vec{\alpha})]|^2\right)$$

Proof Sketch:

Proof of $\frac{1}{2}R_{\mathcal{S}_{\mathrm{Haar}_n}}(\vec{\alpha}) \leq \frac{2^n}{2^n+1}R_{\mathcal{Q}}(\vec{\alpha})$**:**

- n-qubit Haar risk vs squared Hilbert-Schmidt inner product:

$$R_{\mathcal{S}_{\mathrm{Haar_n}}}(\vec{\alpha}) = \frac{2^n}{2^n+1}\left(1 - \frac{1}{4^n}|\operatorname{tr}[U^\dagger V(\vec{\alpha})]|^2\right)$$

- Insert an arbitrary $\tilde{U} \in \mathcal{U}(\mathbb{C})^{2^n}$ (thought of as from ensemble):

Proof Sketch:

Proof of $\frac{1}{2} R_{\mathcal{S}_{\text{Haar}_n}}(\vec{\alpha}) \leq \frac{2^n}{2^n+1} R_{\mathcal{Q}}(\vec{\alpha})$**:**

- n-qubit Haar risk vs squared Hilbert-Schmidt inner product:

$$R_{\mathcal{S}_{\text{Haar}_\text{n}}}(\vec{\alpha}) = \frac{2^n}{2^n+1}\left(1 - \frac{1}{4^n}|\operatorname{tr}[U^\dagger V(\vec{\alpha})]|^2\right)$$

- Insert an arbitrary $\tilde{U} \in \mathcal{U}(\mathbb{C})^{2^n}$ (thought of as from ensemble):

$$1 - \frac{1}{4^n}|\operatorname{tr}[U^\dagger V(\vec{\alpha})]|^2 = 1 - \frac{1}{4^n}|\operatorname{tr}[(\tilde{U}^\dagger U \tilde{U})^\dagger (\tilde{U} V(\vec{\alpha}) \tilde{U}^\dagger)]|^2$$

Proof Sketch:

Proof of $\frac{1}{2}R_{\mathcal{S}_{\text{Haar}_n}}(\vec{\alpha}) \leq \frac{2^n}{2^n+1}R_{\mathcal{Q}}(\vec{\alpha})$**:**

- n-qubit Haar risk vs squared Hilbert-Schmidt inner product:

$$R_{\mathcal{S}_{\text{Haar}_\text{n}}}(\vec{\alpha}) = \frac{2^n}{2^n+1}\left(1 - \frac{1}{4^n}|\operatorname{tr}[U^\dagger V(\vec{\alpha})]|^2\right)$$

- Insert an arbitrary $\tilde{U} \in \mathcal{U}(\mathbb{C})^{2^n}$ (thought of as from ensemble):

$$1 - \frac{1}{4^n}|\operatorname{tr}[U^\dagger V(\vec{\alpha})]|^2 = 1 - \frac{1}{4^n}|\operatorname{tr}[(\tilde{U}^\dagger U \tilde{U})^\dagger (\tilde{U} V(\vec{\alpha}) \tilde{U}^\dagger)]|^2$$

- Squared Hilbert-Schmidt inner product as a Pauli average:

Proof Sketch:

Proof of $\frac{1}{2} R_{\mathcal{S}_{\mathrm{Haar}_n}}(\vec{\alpha}) \leq \frac{2^n}{2^n+1} R_{\mathcal{Q}}(\vec{\alpha})$**:**

- $\boldsymbol{n}$-qubit Haar risk vs squared Hilbert-Schmidt inner product:

$$R_{\mathcal{S}_{\mathrm{Haar}_\mathrm{n}}}(\vec{\alpha}) = \frac{2^n}{2^n+1}\left(1 - \frac{1}{4^n}|\operatorname{tr}[U^\dagger V(\vec{\alpha})]|^2\right)$$

- Insert an arbitrary $\tilde{U} \in \mathcal{U}(\mathbb{C})^{2^n}$ (thought of as from ensemble):

$$1 - \frac{1}{4^n}|\operatorname{tr}[U^\dagger V(\vec{\alpha})]|^2 = 1 - \frac{1}{4^n}|\operatorname{tr}[(\tilde{U}^\dagger U \tilde{U})^\dagger (\tilde{U} V(\vec{\alpha}) \tilde{U}^\dagger)]|^2$$

- Squared Hilbert-Schmidt inner product as a Pauli average:

$$|\operatorname{tr}[A^\dagger B]|^2 = \frac{1}{2^n} \sum_{P \in \{\mathbb{1}, X, Y, Z\}^{\otimes n}} \operatorname{tr}\left[P A^\dagger B P B^\dagger A\right]$$

Proof Sketch:

Proof Sketch:

Proof of $\frac{1}{2} R_{\mathcal{S}_{\mathrm{Haar}_n}}(\vec{\alpha}) \leq \frac{2^n}{2^n+1} R_{\mathcal{Q}}(\vec{\alpha})$ **continued:**

Proof Sketch:

Proof of $\frac{1}{2} R_{\mathcal{S}_{\mathrm{Haar}_n}}(\vec{\alpha}) \leq \frac{2^n}{2^n+1} R_{\mathcal{Q}}(\vec{\alpha})$ continued:

- Writing $W = U^\dagger V(\vec{\alpha})$ and spectral expansion of P give

Proof Sketch:

Proof of $\frac{1}{2} R_{\mathcal{S}_{\mathrm{Haar}_n}}(\vec{\alpha}) \leq \frac{2^n}{2^n+1} R_{\mathcal{Q}}(\vec{\alpha})$ **continued:**

- Writing $W = U^\dagger V(\vec{\alpha})$ and spectral expansion of P give

$$1 - \frac{1}{4^n} |\operatorname{tr}[(\tilde{U}^\dagger U \tilde{U})^\dagger (\tilde{U} V(\vec{\alpha}) \tilde{U}^\dagger)]|^2$$
$$= \mathbb{E}_{|s\rangle \sim \{|0\rangle, |1\rangle, |+\rangle, |-\rangle, |y+\rangle, |y-\rangle\}^{\otimes n}} \mathbb{E}_{P \sim \{\mathbb{1}, X, Y, Z\}^{\otimes n} : \langle s|P|s\rangle \neq 0} \left[1 - \langle s| P |s\rangle \cdot \langle s| \tilde{U}^\dagger W^\dagger \tilde{U} P \tilde{U}^\dagger W \tilde{U} |s\rangle \right]$$

Proof Sketch:

Proof of $\frac{1}{2}R_{\mathcal{S}_{\mathrm{Haar}_n}}(\vec{\alpha}) \leq \frac{2^n}{2^n+1}R_{\mathcal{Q}}(\vec{\alpha})$ **continued:**

- Writing $W = U^\dagger V(\vec{\alpha})$ and spectral expansion of P give

$$1 - \frac{1}{4^n}|\operatorname{tr}[(\tilde{U}^\dagger U\tilde{U})^\dagger(\tilde{U}V(\vec{\alpha})\tilde{U}^\dagger)]|^2$$
$$= \mathbb{E}_{|s\rangle\sim\{|0\rangle,|1\rangle,|+\rangle,|-\rangle,|y+\rangle,|y-\rangle\}^{\otimes n}}\mathbb{E}_{P\sim\{\mathbb{1},X,Y,Z\}^{\otimes n}:\langle s|P|s\rangle\neq 0}\left[1 - \langle s|P|s\rangle\cdot\langle s|\tilde{U}^\dagger W^\dagger\tilde{U}P\tilde{U}^\dagger W\tilde{U}|s\rangle\right]$$

- Bounding the expression inside the expectation:

Proof Sketch:

Proof of $\frac{1}{2} R_{\mathcal{S}_{\mathrm{Haar}_n}}(\vec{\alpha}) \leq \frac{2^n}{2^n+1} R_{\mathcal{Q}}(\vec{\alpha})$ continued:

- Writing $W = U^\dagger V(\vec{\alpha})$ and spectral expansion of P give

$$\begin{aligned}&1 - \frac{1}{4^n} |\operatorname{tr}[(\tilde{U}^\dagger U \tilde{U})^\dagger (\tilde{U} V(\vec{\alpha}) \tilde{U}^\dagger)]|^2 \\ &= \mathbb{E}_{|s\rangle \sim \{|0\rangle, |1\rangle, |+\rangle, |-\rangle, |y+\rangle, |y-\rangle\}^{\otimes n}} \mathbb{E}_{P \sim \{\mathbb{1}, X, Y, Z\}^{\otimes n} : \langle s|P|s\rangle \neq 0} \left[1 - \langle s| P |s\rangle \cdot \langle s| \tilde{U}^\dagger W^\dagger \tilde{U} P \tilde{U}^\dagger W \tilde{U} |s\rangle \right]\end{aligned}$$

- Bounding the expression inside the expectation:

$$0 \leq 1 - \langle s| P |s\rangle \cdot \langle s| W^\dagger P W |s\rangle \leq 2 \left(1 - |\langle s| W |s\rangle|^2 \right)$$

Proof Sketch:

Proof of $\frac{1}{2}R_{\mathcal{S}_{\mathrm{Haar}_n}}(\vec{\alpha}) \leq \frac{2^n}{2^n+1}R_{\mathcal{Q}}(\vec{\alpha})$ **continued:**

- Writing $W = U^\dagger V(\vec{\alpha})$ and spectral expansion of P give

$$1 - \frac{1}{4^n}|\operatorname{tr}[(\tilde{U}^\dagger U\tilde{U})^\dagger(\tilde{U}V(\vec{\alpha})\tilde{U}^\dagger)]|^2$$
$$= \mathbb{E}_{|s\rangle\sim\{|0\rangle,|1\rangle,|+\rangle,|-\rangle,|y+\rangle,|y-\rangle\}^{\otimes n}}\mathbb{E}_{P\sim\{\mathbb{1},X,Y,Z\}^{\otimes n}:\langle s|P|s\rangle\neq 0}\left[1 - \langle s|P|s\rangle\cdot\langle s|\tilde{U}^\dagger W^\dagger\tilde{U}P\tilde{U}^\dagger W\tilde{U}|s\rangle\right]$$

- Bounding the expression inside the expectation:

$$0 \leq 1 - \langle s|P|s\rangle\cdot\langle s|W^\dagger PW|s\rangle \leq 2\left(1 - |\langle s|W|s\rangle|^2\right)$$

- Finishing up (using that random stabilizers form a 2-design and that our ensemble is locally scrambled)

Proof Sketch:

Proof Sketch:

Proof of $\frac{2^n}{2^n+1} R_{\mathcal{Q}}(\vec{\alpha}) \leq R_{\mathcal{S}_{\mathrm{Haar}_n}}(\vec{\alpha})$**:**

Proof Sketch:

Proof of $\frac{2^n}{2^n+1} R_{\mathcal{Q}}(\vec{\alpha}) \leq R_{\mathcal{S}_{\mathrm{Haar}_n}}(\vec{\alpha})$:

- Locally scrambled risk via partial traces:

Proof Sketch:

Proof of $\frac{2^n}{2^n+1} R_{\mathcal{Q}}(\vec{\alpha}) \leq R_{\mathcal{S}_{\mathrm{Haar}_n}}(\vec{\alpha})$**:**

- Locally scrambled risk via partial traces:

$$R_{\mathcal{Q}}(\vec{\alpha}) = 1 - \frac{1}{6^n} \sum_{A\subseteq\{1,\ldots,n\}} \mathbb{E}_{\tilde{U}\sim\mathcal{U}_{\mathrm{test}}} \left[\left\| \mathrm{tr}_{A^c}\left[\tilde{U}^\dagger (U^\dagger V(\vec{\alpha})) \tilde{U} \right] \right\|_F^2 \right]$$

Proof Sketch:

Proof of $\frac{2^n}{2^n+1} R_{\mathcal{Q}}(\vec{\alpha}) \leq R_{\mathcal{S}_{\mathrm{Haar}_n}}(\vec{\alpha})$:

- Locally scrambled risk via partial traces:

$$R_{\mathcal{Q}}(\vec{\alpha}) = 1 - \frac{1}{6^n} \sum_{A\subseteq\{1,\ldots,n\}} \mathbb{E}_{\tilde{U}\sim\mathcal{U}_{\mathrm{test}}} \left[\left\| \mathrm{tr}_{A^c}\left[\tilde{U}^\dagger (U^\dagger V(\vec{\alpha})) \tilde{U} \right] \right\|_F^2 \right]$$

- Controlling Frobenius norms of partial traces:

Proof Sketch:

Proof of $\frac{2^n}{2^n+1} R_{\mathcal{Q}}(\vec{\alpha}) \leq R_{\mathcal{S}_{\mathrm{Haar}_n}}(\vec{\alpha})$:

- Locally scrambled risk via partial traces:

$$R_{\mathcal{Q}}(\vec{\alpha}) = 1 - \frac{1}{6^n} \sum_{A \subseteq \{1,\ldots,n\}} \mathbb{E}_{\tilde{U} \sim \mathcal{U}_{\mathrm{test}}} \left[\left\| \mathrm{tr}_{A^c} \left[\tilde{U}^\dagger (U^\dagger V(\vec{\alpha})) \tilde{U} \right] \right\|_F^2 \right]$$

- Controlling Frobenius norms of partial traces:

$$\left\| \mathrm{tr}_{A^c} \left[\tilde{U}^\dagger W \tilde{U} \right] \right\|_F^2 \geq \frac{1}{2^{|A|}} \left| \mathrm{tr}\left[W \right] \right|^2$$

Proof Sketch:

Proof of $\frac{2^n}{2^n+1} R_{\mathcal{Q}}(\vec{\alpha}) \leq R_{\mathcal{S}_{\mathrm{Haar}_n}}(\vec{\alpha})$**:**

- Locally scrambled risk via partial traces:

$$R_{\mathcal{Q}}(\vec{\alpha}) = 1 - \frac{1}{6^n} \sum_{A \subseteq \{1,\ldots,n\}} \mathbb{E}_{\tilde{U} \sim \mathcal{U}_{\text{test}}} \left[\left\| \mathrm{tr}_{A^c} \left[\tilde{U}^\dagger (U^\dagger V(\vec{\alpha})) \tilde{U} \right] \right\|_F^2 \right]$$

- Controlling Frobenius norms of partial traces:

$$\left\| \mathrm{tr}_{A^c} \left[\tilde{U}^\dagger W \tilde{U} \right] \right\|_F^2 \geq \frac{1}{2^{|A|}} \left| \mathrm{tr}\left[W \right] \right|^2$$

- Finishing up (using again $R_{\mathcal{S}_{\mathrm{Haar}_\mathrm{n}}}(\vec{\alpha}) = \frac{2^n}{2^n+1} \left(1 - \frac{1}{4^n} |\operatorname{tr}[U^\dagger V(\vec{\alpha})]|^2\right)$)

OOD Generalization for QNNs

What OOD generalization bounds we get for unitary learning with QNNs on locally scrambled ensembles

Lifting ID to OOD Generalization

Lifting ID to OOD Generalization

Corollary (Locally scrambled OOD generalization from ID generalization [3]):

[3] **M.C.C.**, H.-Y. Huang, N. Ezzell, J. Gibbs, A.T. Sornborger, L. Cincio, P.J. Coles, Z. Holmes; *arXiv:2204.10268 (2022)*

Lifting ID to OOD Generalization

Corollary (Locally scrambled OOD generalization from ID generalization [3]):

Let $\mathcal{P}, \mathcal{Q} \in \mathbb{S}_{\mathrm{LS}}^{(2)}$. Let U be an unknown n-qubit unitary. Let $V(\vec{\alpha})$ be an n-qubit unitary QNN that is trained using training data $\mathcal{D}_{\mathcal{Q}}(N)$ containing N input-output pairs, with inputs drawn from the ensemble $\mathcal{Q}$. Then, for any parameter setting $\vec{\alpha}$,

$$R_{\mathcal{P}}(\vec{\alpha}) \leq 2\left(C_{\mathcal{D}_{\mathcal{Q}}(N)}(\vec{\alpha}) + \mathrm{gen}_{\mathcal{Q},\mathcal{D}_{\mathcal{Q}}(N)}(\vec{\alpha})\right) .$$

[3] **M.C.C.**, H.-Y. Huang, N. Ezzell, J. Gibbs, A.T. Sornborger, L. Cincio, P.J. Coles, Z. Holmes; *arXiv:2204.10268 (2022)*

Lifting ID to OOD Generalization

Corollary (Locally scrambled OOD generalization from ID generalization [3]):

Let $\mathcal{P}, \mathcal{Q} \in \mathbb{S}_{\mathrm{LS}}^{(2)}$. Let U be an unknown n-qubit unitary. Let $V(\vec{\alpha})$ be an n-qubit unitary QNN that is trained using training data $\mathcal{D}_{\mathcal{Q}}(N)$ containing N input-output pairs, with inputs drawn from the ensemble $\mathcal{Q}$. Then, for any parameter setting $\vec{\alpha}$,

$$R_{\mathcal{P}}(\vec{\alpha}) \leq 2\left(C_{\mathcal{D}_{\mathcal{Q}}(N)}(\vec{\alpha}) + \mathrm{gen}_{\mathcal{Q},\mathcal{D}_{\mathcal{Q}}(N)}(\vec{\alpha})\right).$$

Take-home message:

[3] **M.C.C.**, H.-Y. Huang, N. Ezzell, J. Gibbs, A.T. Sornborger, L. Cincio, P.J. Coles, Z. Holmes; *arXiv:2204.10268 (2022)*

Lifting ID to OOD Generalization

Corollary (Locally scrambled OOD generalization from ID generalization [3]):

Let $\mathcal{P}, \mathcal{Q} \in \mathbb{S}_{\mathrm{LS}}^{(2)}$. Let U be an unknown n-qubit unitary. Let $V(\vec{\alpha})$ be an n-qubit unitary QNN that is trained using training data $\mathcal{D}_{\mathcal{Q}}(N)$ containing N input-output pairs, with inputs drawn from the ensemble $\mathcal{Q}$. Then, for any parameter setting $\vec{\alpha}$,

$$R_{\mathcal{P}}(\vec{\alpha}) \leq 2\left(C_{\mathcal{D}_{\mathcal{Q}}(N)}(\vec{\alpha}) + \mathrm{gen}_{\mathcal{Q},\mathcal{D}_{\mathcal{Q}}(N)}(\vec{\alpha})\right).$$

Take-home message:

OOD risk controlled by training cost and ID generalization error.

[3] **M.C.C.**, H.-Y. Huang, N. Ezzell, J. Gibbs, A.T. Sornborger, L. Cincio, P.J. Coles, Z. Holmes; *arXiv:2204.10268 (2022)*

Locally Scrambled OOD Generalization for Learning Unitaries with QNNs

Locally Scrambled OOD Generalization for Learning Unitaries with QNNs

Consequence of our "lifting corollary":

Locally Scrambled OOD Generalization for Learning Unitaries with QNNs

Consequence of our "lifting corollary":

Any ID generalization bound for QNNs **directly** gives rise to a locally scrambled OOD generalization bound for unitary learning!

Locally Scrambled OOD Generalization for Learning Unitaries with QNNs

Locally Scrambled OOD Generalization for Learning Unitaries with QNNs

Concrete Example, using the ID generalization bound from [4]:

[4] **M.C.C.**, H.-Y. Huang, M. Cerezo, K. Sharma, A.T. Sornborger, L. Cincio, P.J. Coles; Nat Commun 13, 4919 (2022)

Locally Scrambled OOD Generalization for Learning Unitaries with QNNs

Concrete Example, using the ID generalization bound from [4]:

Corollary:

[4] **M.C.C.**, H.-Y. Huang, M. Cerezo, K. Sharma, A.T. Sornborger, L. Cincio, P.J. Coles; Nat Commun 13, 4919 (2022)

Locally Scrambled OOD Generalization for Learning Unitaries with QNNs

Concrete Example, using the ID generalization bound from [4]:

Corollary:
Let $\mathcal{P}, \mathcal{Q} \in \mathbb{S}_{\mathrm{LS}}^{(2)}$. Let $U \in \mathcal{U}\left((\mathbb{C}^2)^{\otimes n}\right)$. Let $V(\vec{\alpha})$ be an n-qubit unitary QNN with T parameterized local gates. When trained with the cost $C_{\mathcal{D}_{\mathcal{Q}}(N)}$ using training data $\mathcal{D}_{\mathcal{Q}}(N)$, the OOD risk w.r.t. $\mathcal{P}$ of the parameter setting $\vec{\alpha}_{\mathrm{opt}}$ after training satisfies, w.h.p. over the choice of training data of size N acc. to $\mathcal{Q}$,

$$R_{\mathcal{P}}(\vec{\alpha}_{\mathrm{opt}}) \leq 2C_{\mathcal{D}_{\mathcal{Q}}(N)}(\vec{\alpha}_{\mathrm{opt}}) + \mathcal{O}\left(\sqrt{\frac{T\log(T)}{N}}\right).$$

[4] **M.C.C.**, H.-Y. Huang, M. Cerezo, K. Sharma, A.T. Sornborger, L. Cincio, P.J. Coles; Nat Commun 13, 4919 (2022)

Applications and Numerics

How OOD generalization can be useful more concretely

Learning a Heisenberg Spin Chain Hamiltonian

Learning a Heisenberg Spin Chain Hamiltonian

- **High-level goal:** Learn unknown parameters in a Hamiltonian from the evolution of product states.

Learning a Heisenberg Spin Chain Hamiltonian

- **High-level goal:** Learn unknown parameters in a Hamiltonian from the evolution of product states.
- **Target Hamiltonian:**

Learning a Heisenberg Spin Chain Hamiltonian

- **High-level goal:** Learn unknown parameters in a Hamiltonian from the evolution of product states.
- **Target Hamiltonian:**

$$H(\vec{p}^*, \vec{q}^*, \vec{r}^*) = \sum_{k=1}^{n-1} \left(Z_k Z_{k+1} + p_k^* X_k X_{k+1}\right) + \sum_{k=1}^{n} \left(q_k^* X_k + r_k^* Z_k\right)$$

Note: We considered the following specific target values

$$p_k^* = \sin\left(\frac{\pi k}{2n}\right) \text{ for } 1 \leq k \leq n-1 \text{ and } q_k^* = \sin\left(\frac{\pi k}{n}\right),\ r_k^* = \cos\left(\frac{\pi k}{n}\right) \text{ for } 1 \leq k \leq n.$$

Learning a Heisenberg Spin Chain Hamiltonian

- **High-level goal:** Learn unknown parameters in a Hamiltonian from the evolution of product states.
- **Target Hamiltonian:**
  $$H(\vec{p}^*, \vec{q}^*, \vec{r}^*) = \sum_{k=1}^{n-1} (Z_k Z_{k+1} + p_k^* X_k X_{k+1}) + \sum_{k=1}^{n} (q_k^* X_k + r_k^* Z_k)$$
- **Ansatz:** $V_L(\vec{p}, \vec{q}, \vec{r}) := (U_{\Delta t}(\vec{p}, \vec{q}, \vec{r}))^L$, with a 2nd order Trotter

Note: We considered the following specific target values

$$p_k^* = \sin\left(\frac{\pi k}{2n}\right) \text{ for } 1 \leq k \leq n-1 \text{ and } q_k^* = \sin\left(\frac{\pi k}{n}\right),\ r_k^* = \cos\left(\frac{\pi k}{n}\right) \text{ for } 1 \leq k \leq n.$$

Learning a Heisenberg Spin Chain Hamiltonian

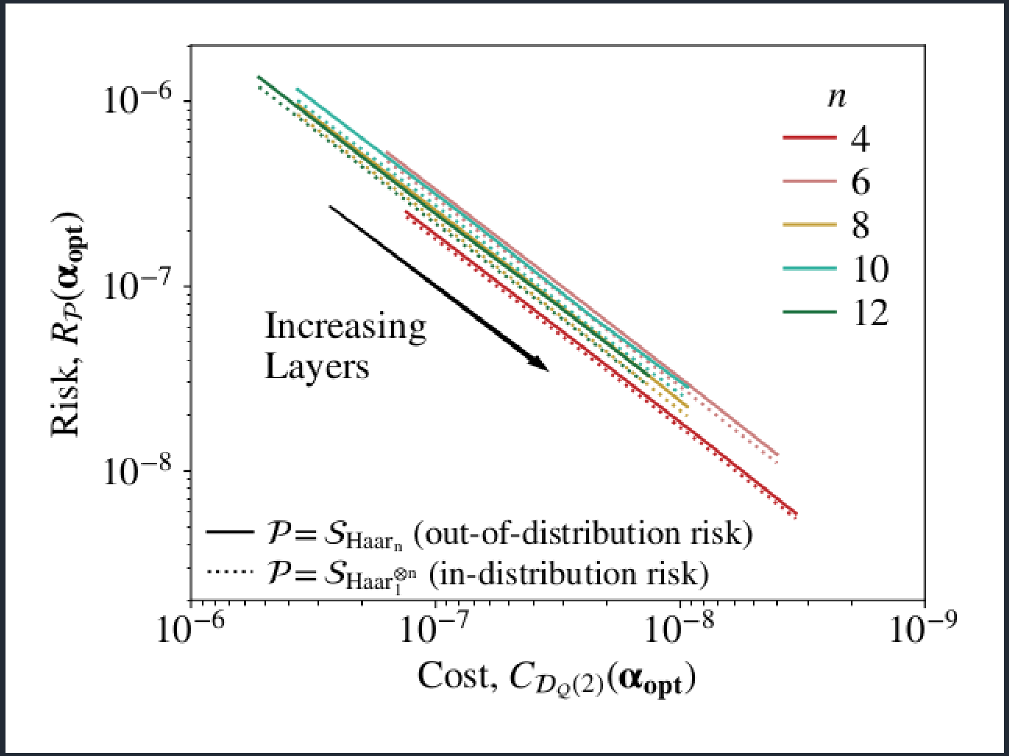
- **High-level goal:** Learn unknown parameters in a Hamiltonian from the evolution of product states.
- **Target Hamiltonian:**
  $$H(\vec{p}^*, \vec{q}^*, \vec{r}^*) = \sum_{k=1}^{n-1} (Z_k Z_{k+1} + p_k^* X_k X_{k+1}) + \sum_{k=1}^{n} (q_k^* X_k + r_k^* Z_k)$$
- **Ansatz:** $V_L(\vec{p}, \vec{q}, \vec{r}) := (U_{\Delta t}(\vec{p}, \vec{q}, \vec{r}))^L$, with a 2nd order Trotter
  $$U_{\Delta t}(\vec{p}, \vec{q}, \vec{r}) = e^{-iH_A(\vec{r})\Delta t/2} e^{-iH_B(\vec{p},\vec{q})\Delta t} e^{-iH_A(\vec{r})\Delta t/2},$$
  where H_A and H_B contain only commuting 2-local terms

Note: We considered the following specific target values

$$p_k^* = \sin\left(\frac{\pi k}{2n}\right) \text{ for } 1 \leq k \leq n-1 \text{ and } q_k^* = \sin\left(\frac{\pi k}{n}\right),\ r_k^* = \cos\left(\frac{\pi k}{n}\right) \text{ for } 1 \leq k \leq n.$$

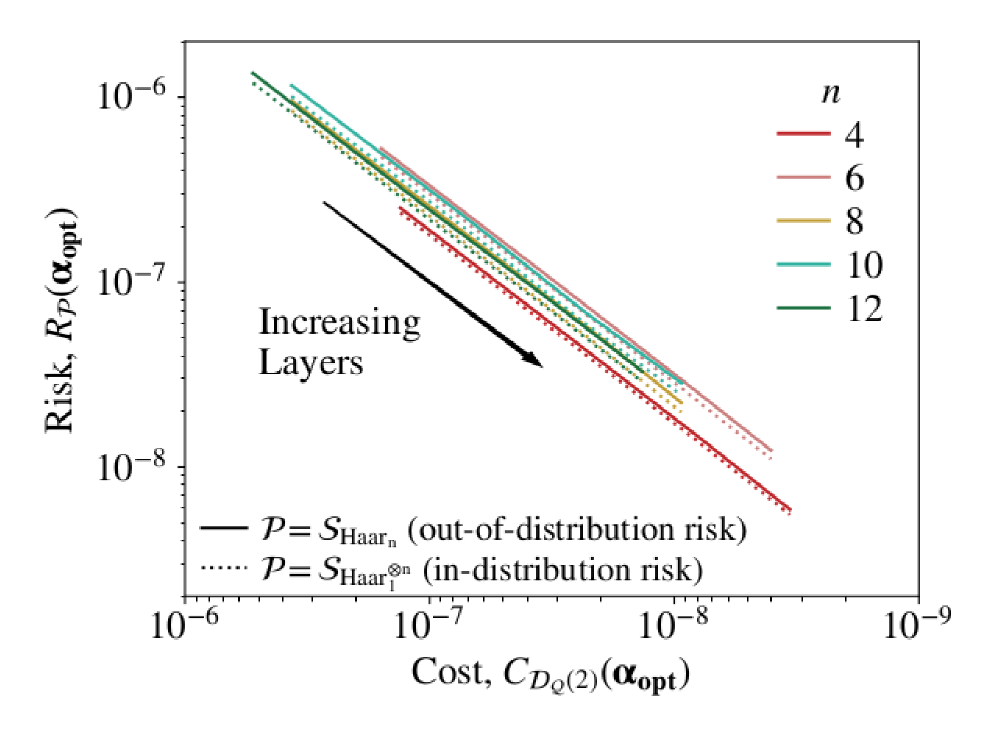
Learning a Heisenberg Spin Chain Hamiltonian

Learning a Heisenberg Spin Chain Hamiltonian

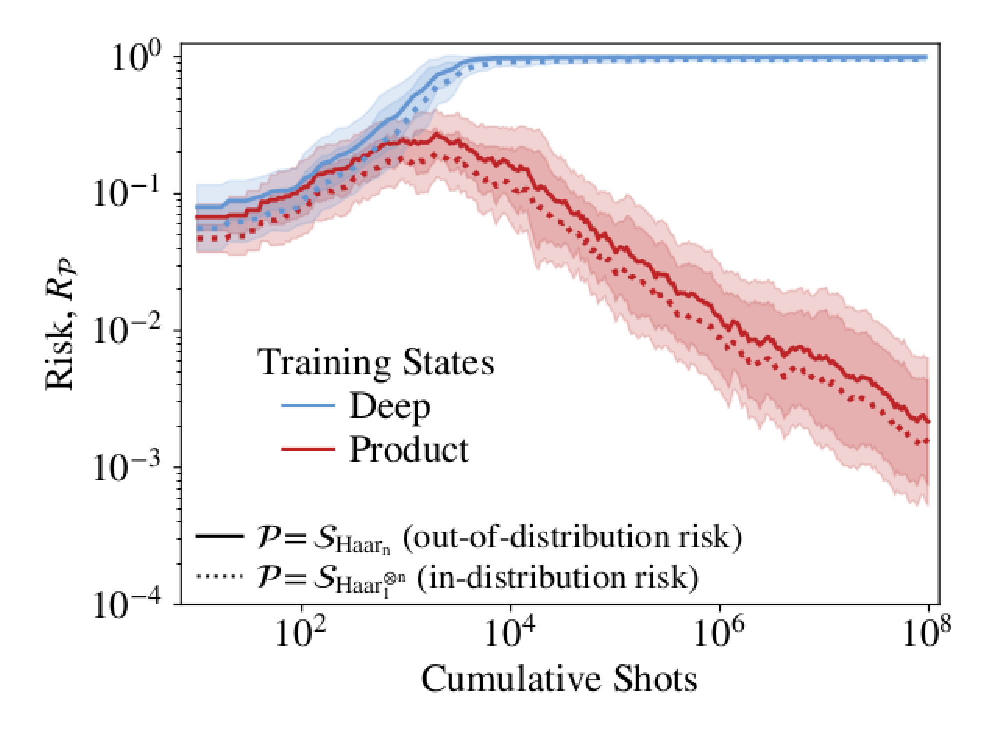


Noise-free simulations

Learning a Heisenberg Spin Chain Hamiltonian



Noise-free simulations



Noisy simulations

Learning a Fast Scrambler

Learning a Fast Scrambler

- **High-level goal:** Learn a fast scrambling unitary.

Learning a Fast Scrambler

- **High-level goal:** Learn a fast scrambling unitary.

- **Target unitary:** $U = \prod_{j=1}^{t} U_j^I U_j^{II}$ with $U_j^I = \prod_{k=1}^{n} u_{j,k}$ and $U_j^{II} = \exp\left(-\frac{i}{2\sqrt{n}} \sum_{k<\ell} Z_k Z_\ell\right)$ [5]

[5] R. Belyansky, P. Bienias, Y.A. Kharkov, A.V. Gorshkov, and B. Swingle; *Phys. Rev. Lett. 125, 130601 (2020)*

Learning a Fast Scrambler

- **High-level goal:** Learn a fast scrambling unitary.

- **Target unitary:** $U = \prod_{j=1}^{t} U_j^I U_j^{II}$ with $U_j^I = \prod_{k=1}^{n} u_{j,k}$ and $U_j^{II} = \exp\left(-\frac{i}{2\sqrt{n}} \sum_{k<\ell} Z_k Z_\ell\right)$ [5]

- **Ansatz:** Same form: $V(\vec{\alpha}) = \prod_{j=1}^{t} V_j^I(\vec{\alpha}_j) U_j^{II}$ with $V_j^I(\vec{\alpha}_j) = \prod_{k=1}^{n} v_{j,k}(\vec{\alpha}_{j,k})$

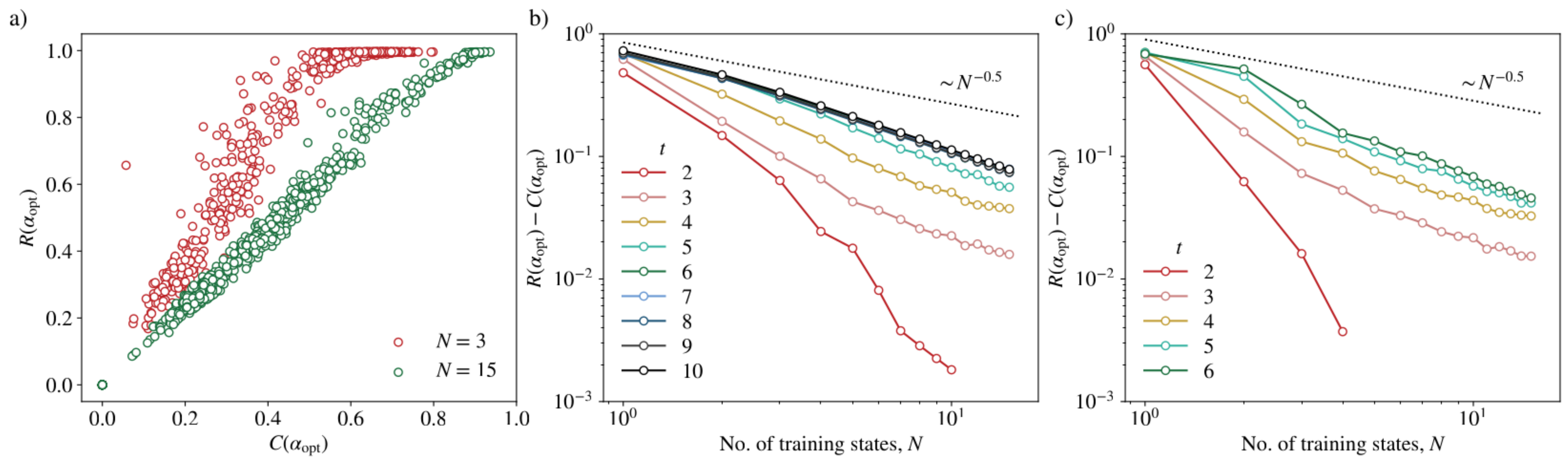
[5] R. Belyansky, P. Bienias, Y.A. Kharkov, A.V. Gorshkov, and B. Swingle; *Phys. Rev. Lett. 125, 130601 (2020)*

Learning a Fast Scrambler

Numerical simulations for 8 qubits:

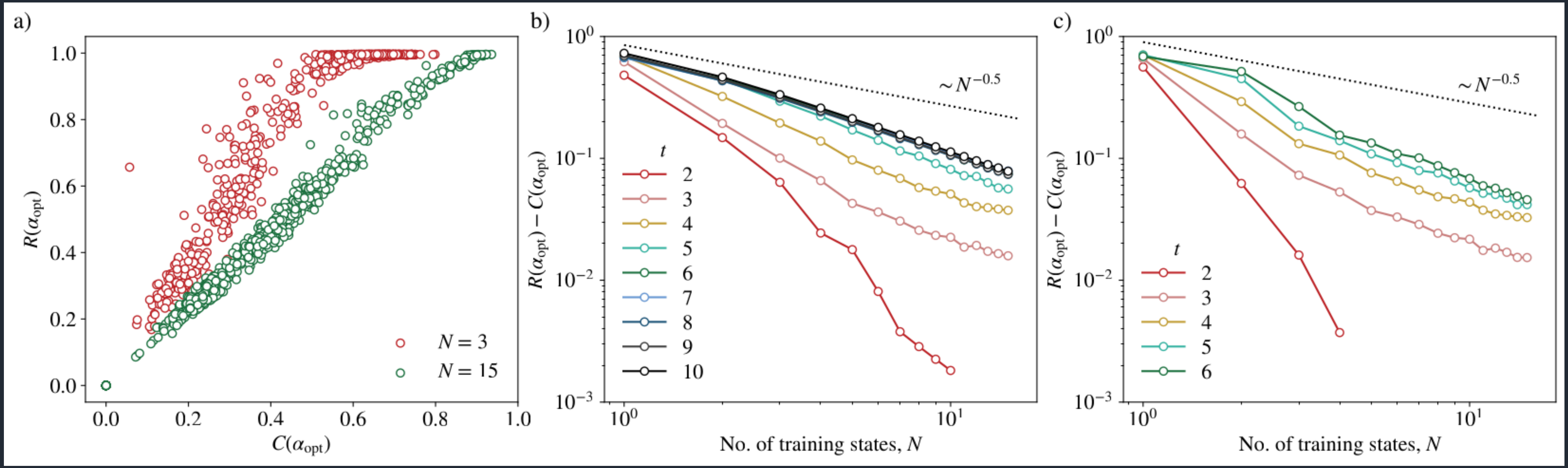
Learning a Fast Scrambler

Numerical simulations for 8 qubits:



Learning a Fast Scrambler

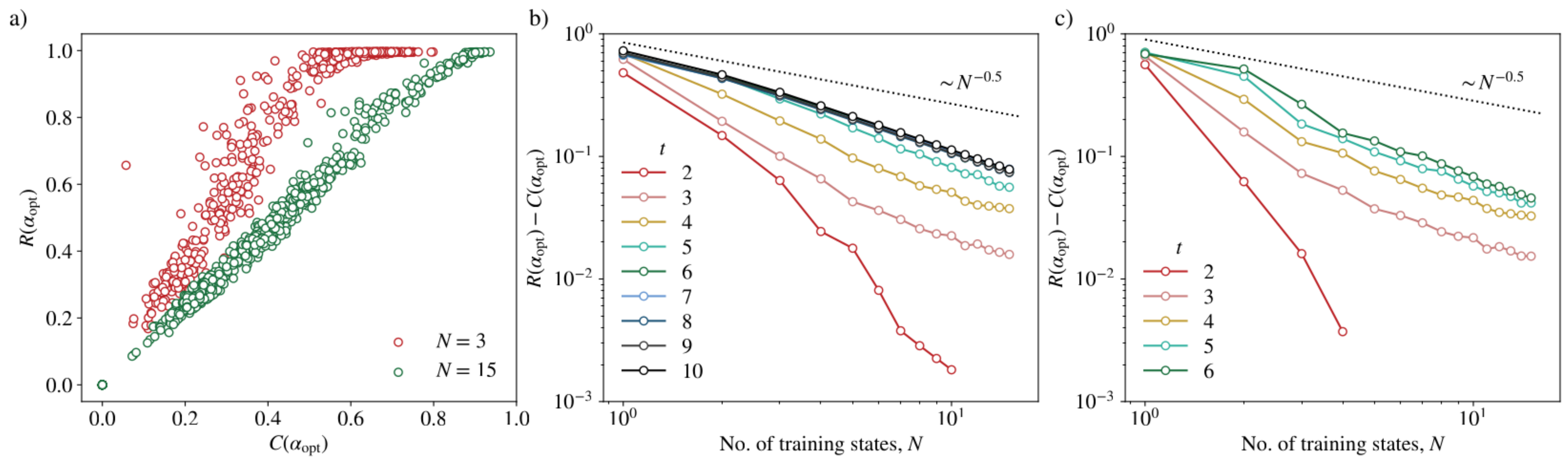
Numerical simulations for 8 qubits:



Testing risk as a function of training cost for $t = 5$

Learning a Fast Scrambler

Numerical simulations for 8 qubits:

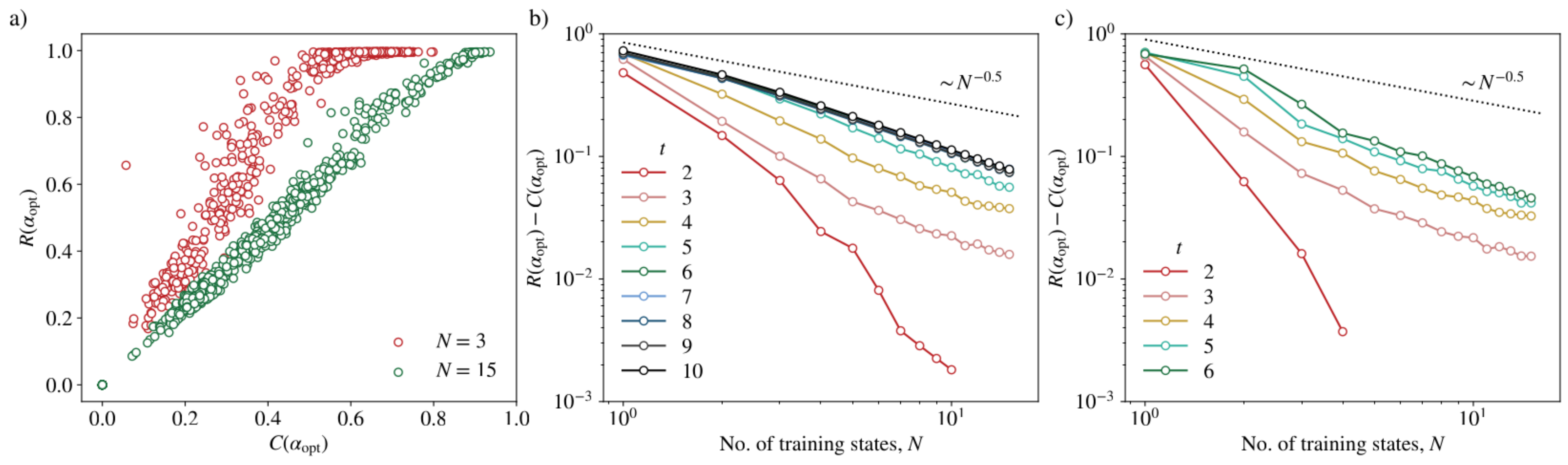


Testing risk as a function of training cost for $t = 5$

Average generalization error versus training data size

Learning a Fast Scrambler

Numerical simulations for 8 qubits:



Testing risk as a function of training cost for $t = 5$

Average generalization error versus training data size

Average generalization error versus training data size, conditioned on successful training

Dynamical Simulation [6]

[6] J. Gibbs, Z. Holmes, **M.C.C.**, N. Ezzell, H.-Y. Huang, L. Cincio, A.T. Sornborger, P.J. Coles; *arXiv:2204.10269 (2022)*

Dynamical Simulation [6]

- **High-level goal:** Efficient procedure for simulating long-time quantum evolutions

[6] J. Gibbs, Z. Holmes, **M.C.C.**, N. Ezzell, H.-Y. Huang, L. Cincio, A.T. Sornborger, P.J. Coles; *arXiv:2204.10269 (2022)*

Dynamical Simulation [6]

- **High-level goal:** Efficient procedure for simulating long-time quantum evolutions
- **High-level idea:** Time-dependent QNN that

[6] J. Gibbs, Z. Holmes, **M.C.C.**, N. Ezzell, H.-Y. Huang, L. Cincio, A.T. Sornborger, P.J. Coles; *arXiv:2204.10269 (2022)*

Dynamical Simulation [6]

- **High-level goal:** Efficient procedure for simulating long-time quantum evolutions
- **High-level idea:** Time-dependent QNN that
  a) learns the short-time-evolution from simple quantum data, and

[6] J. Gibbs, Z. Holmes, **M.C.C.**, N. Ezzell, H.-Y. Huang, L. Cincio, A.T. Sornborger, P.J. Coles; *arXiv:2204.10269 (2022)*

Dynamical Simulation [6]

- **High-level goal:** Efficient procedure for simulating long-time quantum evolutions
- **High-level idea:** Time-dependent QNN that
  a) learns the short-time-evolution from simple quantum data, and
  b) naturally extrapolates to larger times.

[6] J. Gibbs, Z. Holmes, **M.C.C.**, N. Ezzell, H.-Y. Huang, L. Cincio, A.T. Sornborger, P.J. Coles; *arXiv:2204.10269 (2022)*

Dynamical Simulation [6]

- **High-level goal:** Efficient procedure for simulating long-time quantum evolutions
- **High-level idea:** Time-dependent QNN that
  a) learns the short-time-evolution from simple quantum data, and
  b) naturally extrapolates to larger times.
- **Concrete Ansatz:** Diagonalization with time-dependent diagonal:

[6] J. Gibbs, Z. Holmes, **M.C.C.**, N. Ezzell, H.-Y. Huang, L. Cincio, A.T. Sornborger, P.J. Coles; *arXiv:2204.10269 (2022)*

Dynamical Simulation [6]

- **High-level goal:** Efficient procedure for simulating long-time quantum evolutions
- **High-level idea:** Time-dependent QNN that
  a) learns the short-time-evolution from simple quantum data, and
  b) naturally extrapolates to larger times.
- **Concrete Ansatz:** Diagonalization with time-dependent diagonal: $V_t(\vec{\alpha}) = W(\vec{\theta})D_t(\vec{\gamma})W^\dagger(\vec{\theta})$

[6] J. Gibbs, Z. Holmes, **M.C.C.**, N. Ezzell, H.-Y. Huang, L. Cincio, A.T. Sornborger, P.J. Coles; *arXiv:2204.10269 (2022)*

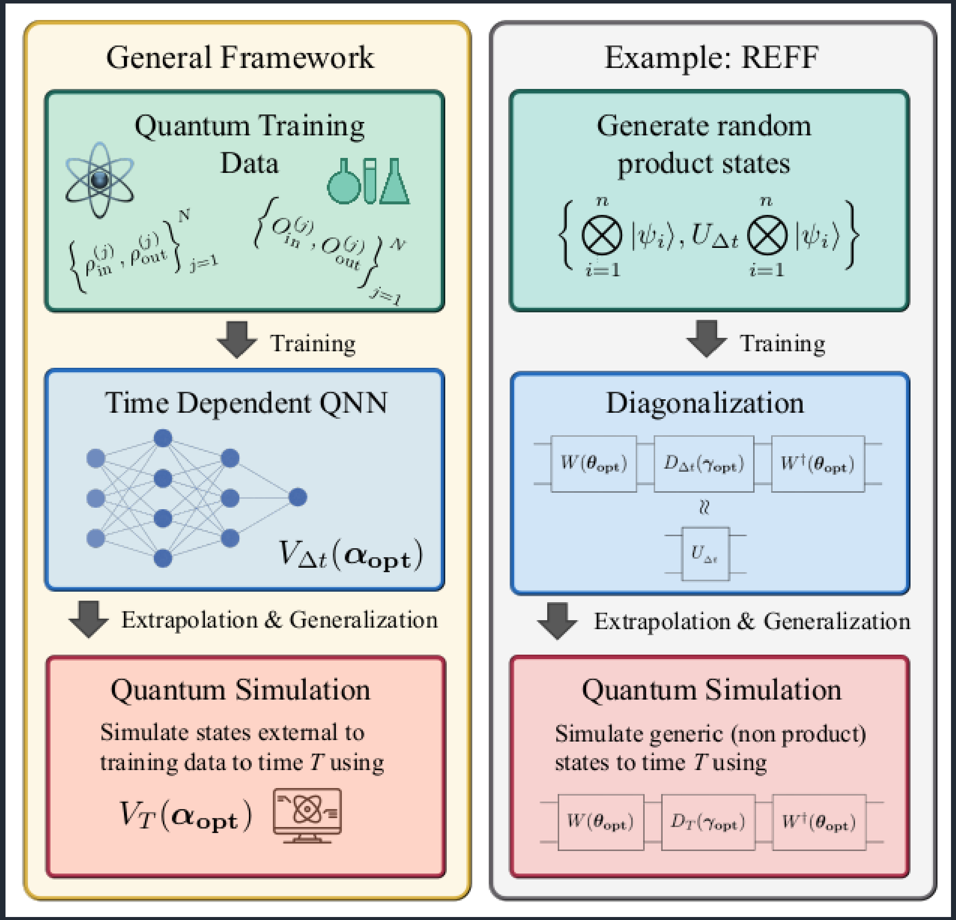
Dynamical Simulation [6]

- **High-level goal:** Efficient procedure for simulating long-time quantum evolutions
- **High-level idea:** Time-dependent QNN that
  a) learns the short-time-evolution from simple quantum data, and
  b) naturally extrapolates to larger times.
- **Concrete Ansatz:** Diagonalization with time-dependent diagonal: $V_t(\vec{\alpha}) = W(\vec{\theta}) D_t(\vec{\gamma}) W^\dagger(\vec{\theta})$

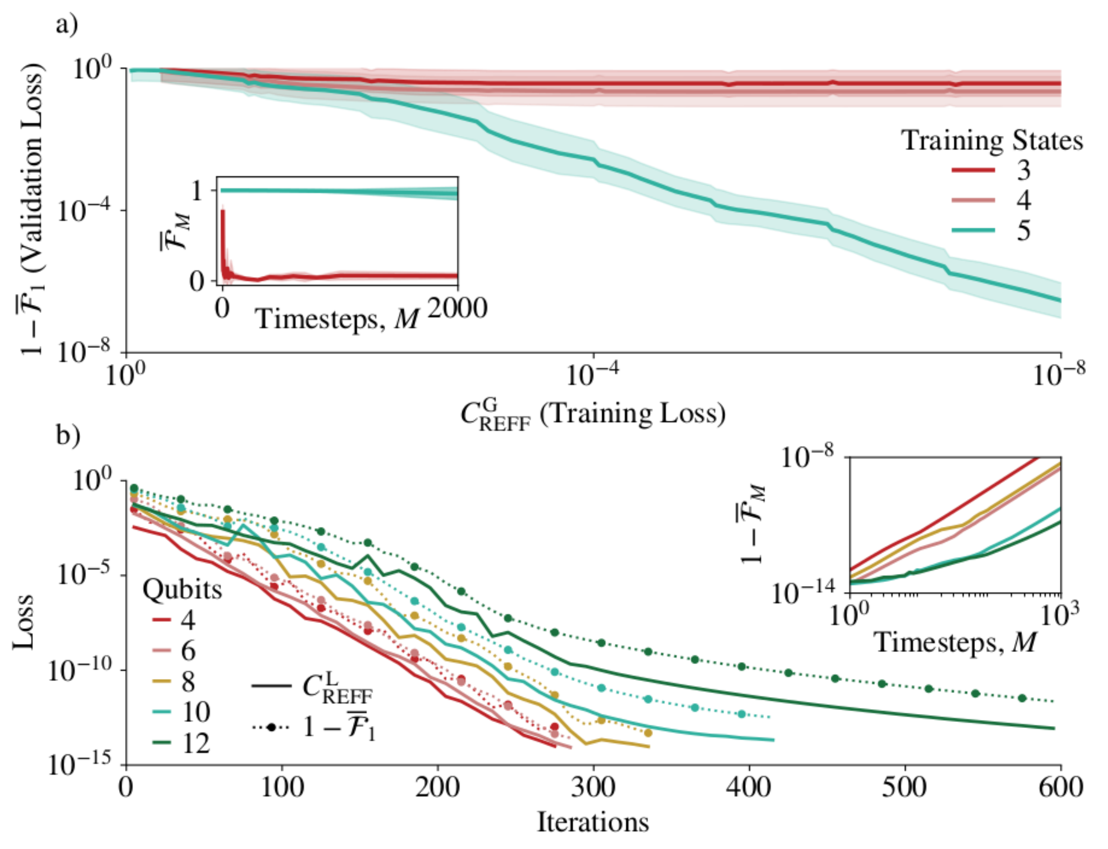
➡ **R**esource-**E**fficient **F**ast **F**orwarding (REFF)

[6] J. Gibbs, Z. Holmes, **M.C.C.**, N. Ezzell, H.-Y. Huang, L. Cincio, A.T. Sornborger, P.J. Coles; *arXiv:2204.10269 (2022)*

Dynamical Simulation [6]

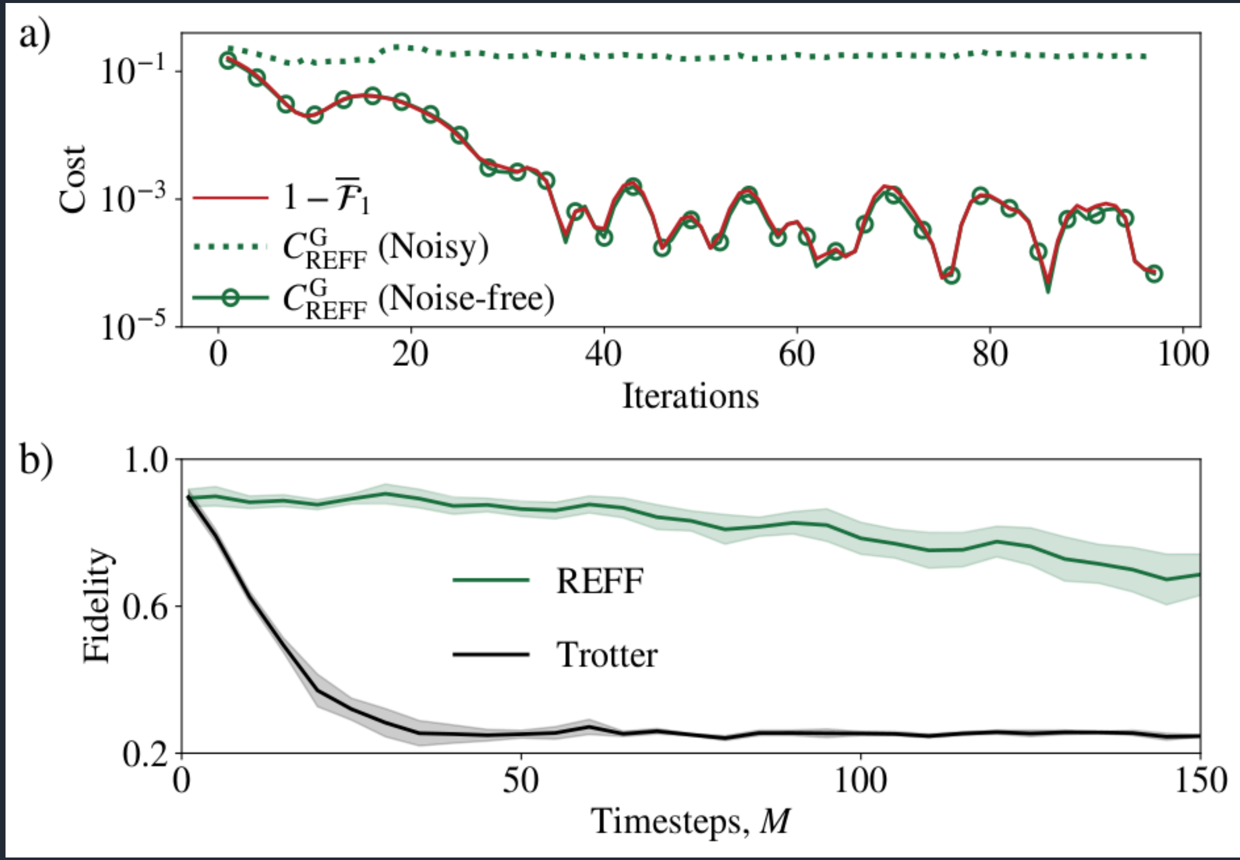


Dynamical Simulation – Simulations [6]



[6] J. Gibbs, Z. Holmes, **M.C.C.**, N. Ezzell, H.-Y. Huang, L. Cincio, A.T. Sornborger, P.J. Coles; *arXiv:2204.10269 (2022)*

Dynamical Simulation – Hardware Implementation [6]



[6] J. Gibbs, Z. Holmes, **M.C.C.**, N. Ezzell, H.-Y. Huang, L. Cincio, A.T. Sornborger, P.J. Coles; *arXiv:2204.10269 (2022)*

Conclusion and Outlook

What we talked about and what one could do next

Conceptual Implications

Conceptual Implications

- **Relevance to NISQ learning of quantum processes:**

Conceptual Implications

- **Relevance to NISQ learning of quantum processes:** NISQ architectures only allow to prepare “simple” states.

Conceptual Implications

- **Relevance to NISQ learning of quantum processes:**

  NISQ architectures only allow to prepare "simple" states.

  ⟶ Our results: "Simple" states can suffice as quantum training data to learn an unknown unitary.

Conceptual Implications

- **Relevance to NISQ learning of quantum processes:**

  NISQ architectures only allow to prepare "simple" states.

  ⟶ Our results: "Simple" states can suffice as quantum training data to learn an unknown unitary.

- **Relevance to classical learning/compiling of unitaries:**

Conceptual Implications

- **Relevance to NISQ learning of quantum processes:**

  NISQ architectures only allow to prepare "simple" states.

  ⟶ Our results: "Simple" states can suffice as quantum training data to learn an unknown unitary.

- **Relevance to classical learning/compiling of unitaries:**

  Tensor network (TN) methods work for low-entangled states.

Conceptual Implications

- **Relevance to NISQ learning of quantum processes:**

  NISQ architectures only allow to prepare "simple" states.

  ⟶ Our results: "Simple" states can suffice as quantum training data to learn an unknown unitary.

- **Relevance to classical learning/compiling of unitaries:**

  Tensor network (TN) methods work for low-entangled states.

  ⟶ Our results: TN methods can learn/compile low-entangling unitaries by training on low-entangled states.

Conceptual Implications

- **Relevance to NISQ learning of quantum processes:**
  NISQ architectures only allow to prepare "simple" states.
  ⟶ Our results: "Simple" states can suffice as quantum training data to learn an unknown unitary.
- **Relevance to classical learning/compiling of unitaries:**
  Tensor network (TN) methods work for low-entangled states.
  ⟶ Our results: TN methods can learn/compile low-entangling unitaries by training on low-entangled states.
- **Physics-inspired ensembles for OOD generalization**

Summary and Open Questions

Summary and Open Questions

Summary

Summary and Open Questions

Summary

- Equivalence of locally scrambled ensembles for unitary learning

Summary and Open Questions

Summary

- Equivalence of locally scrambled ensembles for unitary learning

- Successful unitary learning on “complex” states from training on few “simple” states

Summary and Open Questions

Summary

- Equivalence of locally scrambled ensembles for unitary learning
- Successful unitary learning on "complex" states from training on few "simple" states
- Application to dynamical simulation via REFF

Summary and Open Questions

Summary

- Equivalence of locally scrambled ensembles for unitary learning
- Successful unitary learning on "complex" states from training on few "simple" states
- Application to dynamical simulation via REFF

Open Questions

Summary and Open Questions

Summary

- Equivalence of locally scrambled ensembles for unitary learning
- Successful unitary learning on "complex" states from training on few "simple" states
- Application to dynamical simulation via REFF

Open Questions

- OOD generalization for other QML tasks and data ensembles?

Summary and Open Questions

Summary

- Equivalence of locally scrambled ensembles for unitary learning
- Successful unitary learning on "complex" states from training on few "simple" states
- Application to dynamical simulation via REFF

Open Questions

- OOD generalization for other QML tasks and data ensembles?
- Further applications of OOD generalization in QML?

Summary and Open Questions

Summary

- Equivalence of locally scrambled ensembles for unitary learning
- Successful unitary learning on “complex” states from training on few “simple” states
- Application to dynamical simulation via REFF

Open Questions

- OOD generalization for other QML tasks and data ensembles?
- Further applications of OOD generalization in QML?
- Framework of using QML for near-term quantum simulation?

Your Questions