# APPENDIX

## A.1 Tensor Product and Partial Trace as Matrix Operations

Here we go into more depth on how we construct matrices $W$, $V_y$ and $V_w$ to perform the tensor product and partial trace operations for use in our Algorithm 1.

### A.1.1 Tensor Product

We construct a matrix $W$ that performs tensor product with an $s \times s$ density matrix $\hat{\rho}_B$ with all zeros, except $\hat{\rho}_{1,1} = 1$, i.e., $\hat{\rho}_B = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}_{s \times s}$.

Observe that for an $n \times n$ density matrix $\hat{\rho}_A$, we the tensor product yields an $ns \times ns$ matrix $\hat{\rho}_{AB} = \hat{\rho}_A \otimes \hat{\rho}_B$. Thus, our matrix $W$ will be an $ns \times n$ matrix, such that $\hat{\rho}_A \otimes \hat{\rho}_B = W \hat{\rho}_A W^\dagger$.

To construct $W$, take $n$ of $s \times n$ matrices of zeros, for the $i$th among those $n$ matrices, place '1' on the first row and $i$th column. Then stack all of those matrices vertically to obtain the $ns \times n$ matrix $W$.

Example If we have a $3 \times 3$ density matrix we wished to tensor with a $4 \times 4$ density matrix, we construct $W$ such that:

$$W = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}_{12 \times 3} \tag{1}$$

Then, we find that:

$$\hat{\rho}_A \otimes \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = W \hat{\rho}_A W^\dagger \tag{2}$$

### A.1.2 Partial Trace

The partial trace cannot ordinarily be implemented with a single matrix operation. However, if a projection operator has just been applied, this operation becomes trivial and easy to perform with a matrix multiplication, i.e., $\mathrm{tr}_B \left( \hat{P}_y \hat{\rho}_{AB} \hat{P}_y^\dagger \right) = V_y \hat{P}_y \hat{\rho}_{AB} \hat{P}_y^\dagger V_y^\dagger$. On the other hand, if we wish to take the partial trace without applying a projection operator, i.e., without a measurement of one of the two subsystems, we must take a sum over these matrices like so: $\mathrm{tr}_A (\hat{\rho}_{AB}) = \sum_w V_w \hat{\rho}_{AB} V_w^\dagger$. The subscript of 'tr' tells us which particle we are tracing over.

Partial Trace after Projection Here, we will assume that a projection operator $\hat{P}_y$ corresponding to an observation on the second particle in the same basis was applied on the joint state of a system prior to the partial trace. If this is not the case, we simply construct all matrices $V_y$ for each observation and take a sum as previously described.

The construction of this matrix $V_y$ is straightforward. We take $s$ of $n \times s$ matrices of zeros, and for the $i$th of these $s$ matrices, place '1' on the $y$th column and $i$th row. Then, concatenate these matrices horizontally to obtain $V_y$.

Example If we have a $12 \times 12$ density matrix describing the joint state of a 3-state particle and 4-state particle, we can construct $V_2$ to trace over the second particle after applying a projection operator $\hat{P}_2$ to be:

$$V_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}_{3 \times 12} \tag{3}$$

Then, we find that if we have applied a projection operator:

$$\mathrm{tr}_B \left( \hat{P}_2 \hat{\rho}_{AB} \hat{P}_2^\dagger \right) = V_2 \hat{P}_2 \hat{\rho}_{AB} \hat{P}_2^\dagger V_2^\dagger \tag{4}$$

Partial Trace without Projection Here, we assume that no measurement/projection has been made, since this is how we use it in the algorithm. If this is not the case and there a projection operator was applied, forgo the sum and simply apply the $V_w$ corresponding to the measurement.

To perform partial trace where there has been no observation, we must construct a set of matrices $V_w$, which we apply and then sum over. The construction of each matrix $V_w$ is as follows. We take $s$ of $s \times n$ matrices of zeros, except the $w$th out these $s$ matrices which is an identity matrix. Then concatenate these matrices horizontally to obtain $V_w$.

Example If we have a $12 \times 12$ density matrix describing the joint state of a 3-state particle and 4-state particle, we can construct $V_w$ to trace over the first particle as:

$$V_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{4 \times 12}$$

$$V_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}_{4 \times 12}$$

$$V_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{4 \times 12} \tag{5}$$

Then, we find that:

$$\hat{\rho}_B = \mathrm{tr}_A (\hat{\rho}_{AB}) = \sum_{w=1}^{3} V_w \hat{\rho}_{AB} V_w^\dagger \tag{6}$$

## B.2 Factorizing Unitary Matrices into H Matrices

The proof of this theorem is a generalization of the proof found in Zhao and Jaeger [2007].

**Lemma 1.** For any vector $\vec{x} \in \mathbb{C}^n$ where $n \geq 2$, there exists a matrix $\mathbf{A}$ that is a product of $H$ matrices, such that $\mathbf{A}\vec{x} = \|\vec{x}\|\vec{e}_1$ where $\vec{e}_1 = \begin{bmatrix}1 & 0 & \dots & 0\end{bmatrix}^T_{1\times n}$ (unit vector in $\mathbb{R}^n$).

*Proof.* Consider an arbitrary vector $\vec{x} \in \mathbb{C}^n$, written as $\vec{x} = \begin{bmatrix}x_1 & x_2 & \dots & x_n\end{bmatrix}^T_{1\times n}$. Let us define $y_2 = \sqrt{\|x_1\|^2 + \|x_2\|^2}$ and parameterize the entries $x_1$ and $x_2$ in $\vec{x}$ with $\alpha_2$ and $\beta_2$ so as to write:

$$\begin{aligned} x_1 &= y_2 e^{i\beta_2}\cos(\alpha_2) \\ x_2 &= y_2 e^{i\beta_2}\sin(\alpha_2) \end{aligned} \tag{7}$$

Now consider the action of $\mathbf{H}_1(1, 2, \alpha_2, -2\beta_2, 0, 0)$ on $\vec{x}$:

$$\mathbf{H}_1\vec{x} = \begin{bmatrix} e^{-i\beta_2}\cos\alpha_2 & e^{-i\beta_2}\sin\alpha_2 & 0 & \cdots & 0 \\ -e^{-i\beta_2}\sin\alpha_2 & e^{-i\beta_2}\cos\alpha_2 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} y_2 e^{i\beta_2}\cos(\alpha_2) \\ y_2 e^{i\beta_2}\sin(\alpha_2) \\ x_3 \\ \vdots \\ x_n \end{bmatrix}$$
$$= \begin{bmatrix} y_2 \\ 0 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} \tag{8}$$

Next, we can define $y_3 = \sqrt{\|y_2\|^2 + \|x_3\|^2}$ and parameterize $y_2$ and $x_3$ using $\alpha_3$ and $\beta_3$, just like we previously. We can then apply $\mathbf{H}_2(1, 3, \alpha_3, -2\beta_3, 0, 0)$, and we find that:

$$\mathbf{H_2H_1}\vec{x} = \begin{bmatrix} y_3 \\ 0 \\ 0 \\ x_4 \\ \vdots \\ x_n \end{bmatrix} \tag{9}$$

Following this pattern, we can construct a sequence of $\mathbf{H}$ matrices such that $\mathbf{H}_{n-1}\dots\mathbf{H}_2\mathbf{H}_1\vec{x} = \begin{bmatrix}y_n & 0 & 0 & \dots & 0\end{bmatrix}^T$. Observe that $y_n = \sqrt{\|y_{n-1}\|^2 + \|x_n\|^2} = \sqrt{\|y_{n-2}\|^2 + \|x_{n-1}\|^2 + \|x_n\|^2} = \sqrt{\|x_1\|^2 + \dots \|x_n\|^2} = \|\vec{x}\|$. Thus, with $\mathbf{A} = \mathbf{H}_{n-1}\dots\mathbf{H}_2\mathbf{H}_1$, we have shown that there exists a matrix $\mathbf{A}$ that is a product of $H$ matrices, such that $\mathbf{A}\vec{x} = \|\vec{x}\|\vec{e}_1$. □

**Lemma 2.** Any 2x2 unitary matrix $\mathbf{A}$ can be written as $\mathbf{H}(1, 2, \theta, \phi, \psi, \delta)$.

*Proof.* A generalized 2x2 unitary matrix is written as:

$$\begin{bmatrix} e^{i\phi/2}e^{i\psi}\cos\theta & e^{i\phi/2}e^{i\delta}\sin\theta \\ -e^{i\phi/2}e^{-i\delta}\sin\theta & e^{i\phi/2}e^{-i\psi}\cos\theta \end{bmatrix} \tag{10}$$

which is exactly $\mathbf{H}(1, 2, \theta, \phi, \psi, \delta)$. □

**Theorem 3.** A matrix $\hat{\mathbf{U}}$ is unitary if and only if it can be written as a product of $\mathbf{H}(i, j, \theta, \phi, \psi, \delta)$ matrices with the following form, where $i, j$ denote the rows and columns with special entries:

$\mathbf{H}(i, j, \theta, \phi, \psi, \delta) =$

$$\begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & e^{i\phi/2}e^{i\psi}\cos\theta & \cdots & e^{i\phi/2}e^{i\delta}\sin\theta & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & -e^{i\phi/2}e^{-i\delta}\sin\theta & \cdots & e^{i\phi/2}e^{-i\psi}\cos\theta & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} \tag{11}$$

*Proof.* We will prove both the forward and reverse directions:

1. *A matrix $\hat{\mathbf{U}}$ is unitary if it can be written as a product of $\mathbf{H}$ matrices.*

   Observe that matrix $\mathbf{H}$ is unitary, since $\mathbf{H}^\dagger\mathbf{H} = \mathbb{I}$. A product of unitary matrices is itself unitary, hence a matrix $\hat{\mathbf{U}}$ that is a product of these $\mathbf{H}$ matrices is unitary.

2. *If a matrix $\hat{\mathbf{U}}$ is unitary, it can be written as a product of $\mathbf{H}$ matrices.*

   We will give a proof by induction. We want to show that any $n \times n$ $\hat{\mathbf{U}}$ unitary matrix can be written as product of $\mathbf{H}$ matrices.

   **Base Case** When $n = 2$, i.e., for a $2 \times 2$ unitary matrix, we know that it can be written as product of $\mathbf{H}$ matrices from Lemma 2.

   **Inductive Hypothesis** Assume that the claim holds for $n = k$, i.e., any $k \times k$ unitary matrix can be written as a product of $\mathbf{H}$ matrices.

   With $n = k+1$, consider an arbitrary $(k+1) \times (k+1)$ unitary matrix $\hat{\mathbf{U}} = \begin{bmatrix}\vec{u}_1 & \vec{u}_2 & \dots & \vec{u}_{k+1}\end{bmatrix}$ where $\vec{u}_i$ is the $i$th column. Since $\hat{\mathbf{U}}$ is unitary, $\|\vec{u}_i\| = 1$ for $1 \leq i \leq k+1$. Then, by Lemma 1, we have a matrix $\mathbf{A}$ that is a product of $\mathbf{H}$ matrices such that $\mathbf{A}\vec{u_1} = \|\vec{u}_1\|\vec{e}_1 = \vec{e}_1$.

   Using this matrix, we find that $\hat{\mathbf{U}}' = \mathbf{A}\hat{\mathbf{U}} = \begin{bmatrix}1 & \vec{C} \\ \vec{0} & \mathbf{V}\end{bmatrix}$ where $\vec{0}$ represents a $k \times 1$ column vector, $\vec{C}$ represents a $1 \times k$ row vector, and $\mathbf{V}$ represents a $k \times k$ matrix. But $\hat{\mathbf{U}}'$ is unitary, so $\hat{\mathbf{U}}'(\hat{\mathbf{U}}')^\dagger = \mathbb{I}$, which means $\mathbf{V}\mathbf{V}^\dagger = \mathbb{I}_{k\times k}$ and $\vec{C} = \vec{0}$.

   **Inductive Step** From the inductive hypothesis, we know that $\mathbf{V}$ can be written as a product of $\mathbf{H}$ matrices, so let us write $\mathbf{V} = \mathbf{H}_k, \dots, \mathbf{H}_1$. Next, we take each of these $k \times k$ $\mathbf{H}$ matrices and pad them to obtain $(k+1) \times (k+1)$ matrices $\mathbf{H}'_i = \begin{bmatrix}1 & 0 \\ 0 & \mathbf{H}_i\end{bmatrix}$. Then, we see that $\mathbf{H}'_k, \dots, \mathbf{H}'_1 = \begin{bmatrix}1 & \vec{0} \\ \vec{0} & \mathbf{V}\end{bmatrix} = \hat{\mathbf{U}}'$.

Finally, we can write our arbitrary unitary matrix $\hat{\mathbf{U}} = \mathbf{A}^{-1}\mathbf{H}'_k, \ldots, \mathbf{H}'_1$, which is indeed a product of $\mathbf{H}$ matrices. Hence, we have shown that any unitary matrix can be written as a product of $\mathbf{H}$ matrices.

□

### C.3 Monras et al. [2010] 2-state HQMM

Monras et al. [2010] present a 4-state HMM with a loose lower bound requirement of 3 classical latent states that can be modeled by the following 2-state, 4-output HQMM:

$$\hat{K}_{1,1} = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 \end{pmatrix} \qquad \hat{K}_{1,2} = \begin{pmatrix} 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} \end{pmatrix} \tag{12}$$

$$\hat{K}_{1,3} = \begin{pmatrix} \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \\ \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \end{pmatrix} \qquad \hat{K}_{1,4} = \begin{pmatrix} \frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} \\ -\frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \end{pmatrix} \tag{13}$$

Observe that this model also treats state evolution as unitary since there is only 1 Kraus operator per observable. We use this model to generate 20 training sequences of length 3000, and 10 validation sequences of length 3000, with a 'burn-in' of 1000 to disregard the influence of the starting distribution. Our results in Table 4 show that our algorithm is capable of learning an HQMM that can match the performance of the original model, while the HMM needs more states to model this process.

### D.4 Synthetic Data from a hand-written HMM

We have shown that we can generate data using HQMMs that classical HMMs with the same number of hidden states or parameters struggle to model. In this section, we explore how well HQMMs can model data generated by a classical HMM. In general, randomly generated HMMs generate data that's hard to predict (i.e., DA closer to 0), so we hand-author an arbitrary, well-behaved HMM with full-rank transition matrix $\mathbf{A}$ and full-rank emission matrix $\mathbf{C}$ to compare HQMM learning with EM for HMMs.

$$\mathbf{A} = \begin{bmatrix} 0.8 & 0.01 & 0 & 0.1 & 0.3 & 0 \\ 0.02 & 0.02 & 0.1 & 0.15 & 0.05 & 0 \\ 0.08 & 0.03 & 0.1 & 0.4 & 0.05 & 0.5 \\ 0.05 & 0.04 & 0.5 & 0.35 & 0 & 0.5 \\ 0.03 & 0.5 & 0.03 & 0 & 0.6 & 0 \\ 0.02 & 0.4 & 0.27 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 0.2 & 0 & 0.05 & 0.95 & 0.01 & 0.05 \\ 0.7 & 0.1 & 0.05 & 0.01 & 0.05 & 0.05 \\ 0.05 & 0.8 & 0.1 & 0.02 & 0.05 & 0.04 \\ 0.04 & 0.04 & 0.02 & 0 & 0.84 & 0.11 \\ 0.01 & 0.03 & 0.7 & 0.01 & 0.02 & 0.2 \\ 0 & 0.03 & 0.08 & 0.01 & 0.03 & 0.55 \end{bmatrix} \tag{14}$$

As before, we generate 20 training sequences of length 3000, and 10 validation sequences of length 3000, with a 'burn-in' of 1000 to disregard the influence of the starting distribution. Results are presented in Table 5.

### E.5 Converting Column Stochastic Matrices to Unitary Matrices

Refer to Algorithm 3 on the following page.

**Algorithm 3** $s \times n$ Column-Stochastic Matrix to $ns \times ns$ Unitary Matrix

**Input:** $s \times n$ Column-Stochastic Matrix $\mathbf{A}$

**Output:** $ns \times ns$ block diagonal Unitary Matrix $\hat{U}$ with $n$ blocks of $s \times s$ unitary matrices, zeros everywhere else

1: **Construct an $s \times s$ unitary matrix from each column of $A$**: Let $c_i$ denote the $i$th column of $\mathbf{A}$. First create an $s \times s$ matrix whose each row is the square root of column $c_i$. Find the null space of this matrix, and you will get the $s-1$ vectors that are linearly independent of $c_i$. Make $c_i$ the first column, and the remaining $s-1$ vectors the other columns of an $s \times s$ matrix.

2: **Stack each $s \times s$ matrix on a diagonal**: Follow step 1 for each column of $A$, and obtain $n$ unitary matrices of dimension $s \times s$. Create a block diagonal matrix with each of these smaller unitary matrices along the diagonal, and you will obtain an $ns \times ns$ dimensional unitary matrix $\hat{U}$.

3: **Note: The unitary operator constructed here is designed to be applied on a density matrix tensored with an ancilla density matrix prepared with zeros everywhere except $\hat{\rho}_{1,1} = 1$.**

Table 4: Performance of various HQMMs and HMMs on data generated by the Monras et al. [2010] model. HQMM parameters are given as $(n, s, w)$ and HMM parameters are given as $(n, s)$, where $n$ is the number of hidden states, $s$ is the number of observables, and $w$ is the number of Kraus operators per observable

| Model | P | Train DA | Test DA |
|---|---|---|---|
| $2,4,1-$HQMM (true) | 16 | 0.2505 (0.0037) | 0.2516 (0.0063) |
| $2,4,1-$HQMM (learned) | 16 | 0.2501 (0.0085) | 0.2512 (0.0064) |
| $2,4,2-$HQMM (learned) | 32 | 0.2499 (0.0035) | 0.2508 (0.0060) |
| $2,4-$HMM (learned) | 12 | 0.0960 (0.0085) | 0.0963 (0.0064) |
| $3,4-$HMM (learned) | 21 | 0.1387 (0.0067) | 0.1416 (0.0070) |
| $4,4-$HMM (learned) | 32 | 0.2504 (0.0037) | 0.2515 (0.0062) |

Table 5: Performance of various HQMMs and HMMs on synthetic data generated by an HMM. HQMM parameters are given as $(n, s, w)$ and HMM parameters are given as $(n, s)$, where $n$ is the number of hidden states, $s$ is the number of observables, and $w$ is the number of Kraus operators per observable

| Model | P | Train DA | Test DA |
|---|---|---|---|
| $6,6-$HMM (true) | 72 | 0.1838 (0.0095) | 0.1903 (0.0071) |
| $2,6,1-$HQMM (learned) | 24 | 0.1597 (0.0088) | 0.1659 (0.0073) |
| $3,6,1-$HQMM (learned) | 54 | 0.1655 (0.0101) | 0.1715 (0.0085) |
| $4,6,1-$HQMM (learned) | 96 | 0.1732 (0.0103) | 0.1772 (0.0103) |
| $5,6,1-$HQMM (learned) | 150 | 0.1680 (0.0093) | 0.1706 (0.0084) |
| $5,6,2-$HQMM (learned) | 300 | 0.1817 (0.0096) | 0.1863 (0.0069) |
| $5,6,3-$HQMM (learned) | 450 | 0.1817 (0.0093) | 0.1866 (0.0064) |
| $5,6,5-$HQMM (learned) | 750 | 0.1821 (0.0095) | 0.1877 (0.0060) |
| $6,6,1-$HQMM (learned) | 216 | 0.1713 (0.0113) | 0.1708 (0.0079) |
| $6,6,2-$HQMM (learned) | 432 | 0.1817 (0.0096) | 0.1870 (0.0070) |
| $2,6-$HMM (learned) | 16 | 0.1282 (0.0074) | 0.1314 (0.0062) |
| $3,6-$HMM (learned) | 27 | 0.1555 (0.0097) | 0.1625 (0.0073) |
| $4,6-$HMM (learned) | 40 | 0.1667 (0.0099) | 0.1732 (0.0068) |
| $5,6-$HMM (learned) | 55 | 0.1751 (0.0097 | 0.1816 (0.0070) |
| $6,6-$HMM (learned) | 72 | 0.1841 (0.0095) | 0.1901 (0.0070) |