physical sciences forum

MDPI

Proceeding Paper

Quantum Finite Automata and Quiver Algebras [†]

George Jeffreys * and Siu-Cheong Lau

Department of Mathematics and Statistics, Boston University, 111 Cummington Mall, Boston, MA 02215, USA
* Correspondence: g.jeffreys@northeastern.edu

† Presented at the 41st International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering, Paris, France, 18–22 July 2022.

Abstract: We find an application in quantum finite automata for the ideas and results of [JL21] and [JL22]. We reformulate quantum finite automata with multiple-time measurements using the algebraic notion of a near-ring. This gives a unified understanding towards quantum computing and deep learning. When the near-ring comes from a quiver, we have a nice moduli space of computing machines with a metric that can be optimized by gradient descent.

Keywords: quantum computing; measurement problem; quantum finite automata (QFA); machine learning; near-ring; quiver representations; deep learning; moduli space; noncommutative geometry

check for updates

Citation: Jeffreys, G.; Lau, S.-C. Quantum Finite Automata and Quiver Algebras. *Phys. Sci. Forum* **2022**, *5*, 32. https://doi.org/10.3390/psf2022005032

Academic Editors: Frédéric Barbaresco, Ali Mohammad-Djafari, Frank Nielsen and Martino Trassinelli

Published: 14 December 2022

1. Motivation: QFA and Near-Ring

In quantum theory, the evolution of states is unitary. An observable is modeled by a self-adjoint operator whose eigenvalues are the possible output values, and whose eigenvectors form an orthonormal basis of the state space. As a result, a typical quantum model simply consists of linear operators that form an algebra.

However, when passing from the quantum world to the real world, an actual probabilistic projection to an eigenstate is necessary. Such a probabilistic operation destroys the linear structure, so we need a non-linear (meaning non-distributive) algebraic structure to accommodate such operators. In the 20th century, there were several attempts to solve this problem. See for instance [1,2], and [3] (Chapter 3) for a nice survey. In particular, Pascual Jordan attempted to use near-ring for quantum mechanics.

Let us consider the scenario of quantum computing.

Definition 1 ([4])**.** *A quantum finite automata (QFA) is a tuple $\mathcal{Q} = (V, q_0, F, \Sigma, (U_\sigma)_{\sigma\in\Sigma})$ where:*

1. *V is a finite set of states which generate the Hilbert space $\mathcal{H}_V$;*
2. *$F \subset V$ is a set of final or accept states;*
3. *q_0 the initial state which is a unit vector in $\mathcal{H}_V$;*
4. *Σ is a finite set called the alphabet;*
5. *For each $\sigma \in \Sigma$, U_σ is a unitary operator on $\mathcal{H}_V$.*

An input to a QFA consists of a word w in the alphabet Σ of the form $w = w_1 w_2 \dots w_n$ where $w_i \in \Sigma$ for all i. w acts on the initial state of the QFA by $\langle q_0 | U_w$, where U_w is the matrix $U_w := U_{w_1} U_{w_2} \dots U_{w_n}$, and $\langle q_0|$ is the row vector presentation of q_0. The probability that word w will end in an accept state is

$$Pr(w) = \| \langle q_0 | U_w P \|^2$$

where $P : \mathcal{H}_V \to \mathcal{H}_F$ is the projection from $\mathcal{H}_V$ to subspace $\mathcal{H}_F$ spanned by F.

Please note that the above definition has not taken the probabilistic projection into account. We make the following reformulation.

Definition 2. *A quantum computing machine is a tuple* $((\mathcal{H}_V, h), \mathcal{H}_F, e, \rho_G, \sigma^F)$, *where*

1. $(\mathcal{H}_V, h)$ *is a Hermitian vector space;*
2. $\mathcal{H}_F = \mathbb{C}^n$ *equipped with the standard metric, which is called the framing space;*
3. $e : \mathcal{H}_F \to \mathcal{H}_V$ *is an isometric embedding;*
4. $\rho_G : G \to U(\mathcal{H}_V, h)$ *is a unitary representation of a group G.*
5. $\sigma^F : \mathcal{H}_F \to \mathcal{H}_F$ *is a probabilistic projection.*

Ignoring the last item (5) for the moment, this coincides with Definition 1 by setting $G = \langle \Sigma \rangle$, the free group generated by a set Σ, and fixing an initial vector $q_0 \in \mathcal{H}_V$.

Here, we treat $\mathcal{H}_F$ as a vector space of its own and take an isometric embedding $e : \mathcal{H}_F \to \mathcal{H}_V$, rather than directly identifying $\mathcal{H}_F$ as a subspace in $\mathcal{H}_V$. The state space $\mathcal{H}_V$ is treated as an abstract vector space without a preferred basis, while $\mathcal{H}_F$ is equipped with a fixed basis that has a real physical meaning (like up/down spinning of an electron). The framing map $e : \mathcal{H}_F \to \mathcal{H}_V$ is interpreted as a bridge between the classical and the quantum world; the image of the fixed basis under e determines a subset of pure state vectors of a certain observable. The adjoint $e^* : \mathcal{H}_V \to \mathcal{H}_F$ is an orthogonal projection. In the next section, e is no longer required to be an embedding when we consider *non-unitary generalizations for machine learning*.

For the last item (5), the probabilistic projection $\sigma^F : \mathcal{H}_F \to \mathcal{H}_F$ can be modeled by a probability space. Specifically, consider a $\mathcal{H}_F$-family of random variables

$$k : \mathcal{H}_F \times \Omega \to \{1, \ldots, |F|\}$$

where Ω is a probability space (that has a probability measure), with the assumption that $\Pr(k(v) = j) = \langle \frac{v}{\|v\|}, \epsilon_j \rangle$ for every $v \in \mathcal{H}_F$, where $\epsilon_j \in \mathcal{H}_F$ denotes the j-th basic vector. Then $\sigma^F(v) := \epsilon_{k(v)}$.

The major additional ingredients in Definition 2, compared to Definition 1, are e, e^* and σ^F. Please note that they are not yet included in the machine language, which is currently the group G. Since e and σ^F are not invertible, we cannot enlarge G to include e nor σ^F as a group.

To remedy this, first note that (4) can be replaced with an *algebra* rather than a group, which exhibits linearity and allows not being invertible. Specifically, we require instead:

(4′) $\rho_A : A \to \mathrm{End}(\mathcal{H}_V)$ is an algebra homomorphism for an algebra A (with unit 1_A).

For instance, A can be the free algebra generated by a set Σ.

With such a modification, we can easily include the framing e and e^* into our language by taking the augmented algebra

$$\mathcal{A} = A\langle 1_F, e, e^* \rangle / R \tag{1}$$

where R is generated by the relations $1_F \cdot 1_F = 1_F, 1_F \cdot e^* = e^*, e \cdot 1_F = e, 1_A \cdot e = e, e \cdot e = 0, e^* \cdot e^* = 0, 1_F \cdot e = 0, 1_F \cdot a = 0, e \cdot 1_A = 0, e^* \cdot 1_F = 0, 1_A \cdot e^* = 0$ for any $a \in A$. The unit of $\mathcal{A}$ is $1_A + 1_F$.

However, we cannot further enlarge $\mathcal{A}$ to include σ^F as an algebra. The reason for this is that σ^F always maps to unit vectors and cannot be linear:

$$\sigma^F(v + w) \neq \sigma^F(v) + \sigma^F(w).$$

To extend $\mathcal{A}$ by σ^F which models actual quantum measurement, we need the notion of a *near-ring*. It is a set A with two binary operations $+, \circ$ such that A is a group under '$+$', '$\circ$' is associative, and right multiplication is distributive over addition: $(x + y) \circ z = x \circ z + y \circ z$ for all $x, y, z \in A$ (but left multiplication is not required distributive: $z \circ (x + y) \neq z \circ x + z \circ y$).

Define $\widetilde{\mathcal{A}}$ to be the near-ring

$$\widetilde{\mathcal{A}} := (1_F + e^* \cdot \mathcal{A} \cdot e)\{\sigma^F\}.$$

This near-ring can be understood as the language that controls quantum computing machines. Elements of $\widetilde{\mathcal{A}}$ can be recorded as rooted trees. An example is $a_1\sigma^F \circ (a_{11} + a_{12})$ where $a_1, a_{11}, a_{12} \in (1_F + e^* \cdot \mathcal{A} \cdot e)$. See also the tree on the left-hand side of Figure 1.

The advantage of putting all the algebraic structures into a single near-ring is that we can consider all the quantum computing machines (mathematically $\tilde{\mathcal{A}}$-modules) controlled by a single near-ring at the same time. An element of $\tilde{\mathcal{A}}$ is a quantum algorithm, which can run in all quantum computers controlled by $\tilde{\mathcal{A}}$.

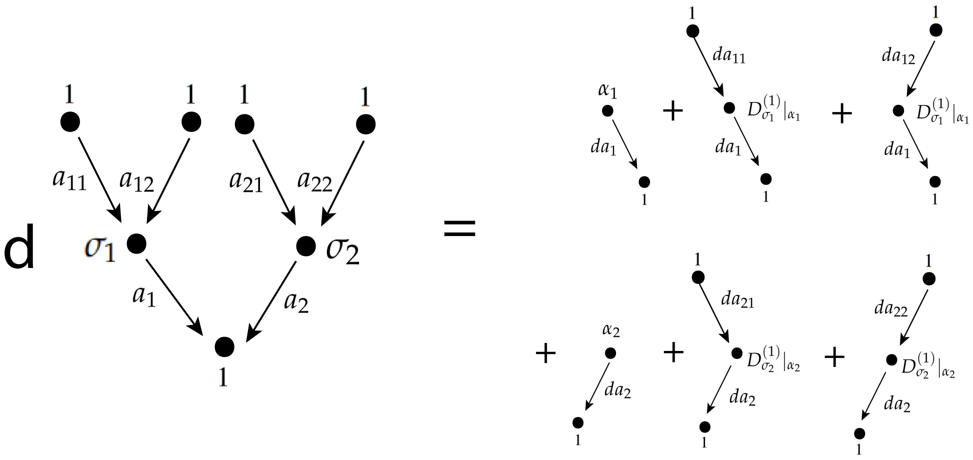

Figure 1. An example of the backpropagation algorithm for the network in Figure 2 when $s = 2$.

2. Near-Ring and Differential Forms

In the setting of Definition 1 and Example 1, it is natural to relax the representations from unitary groups to matrix algebras $\mathfrak{gl}(n, \mathbb{C})$. Moreover, the quantum measurement can also be simulated by a non-linear function (called activation function). Such a modification will produce a computational model of deep learning.

Definition 3. *An activation module consists of:*

1. *A noncommutative algebra A and vector space V, F;*
2. *A family of metrics $h_{(\rho,e)}$ on V over the space of framed A-modules*

$$R = Hom_{alg}(A, End(V)) \times Hom(F, V)$$

which is G-equivariant where $G = \mathrm{GL}(V)$;

3. *A collection of possibly non-linear functions*

$$\sigma_j^F : F \to F.$$

As in (1), we take the augmented algebra $\mathcal{A} = A\langle 1_F, e, e^*\rangle / R$ which produces linear computations in all framed A-modules simultaneously. With item (3), elements in the near-ring

$$\tilde{\mathcal{A}} := (1_F + e^* \cdot \mathcal{A} \cdot e)\{\sigma_1^F, \ldots, \sigma_N^F\}$$

induce non-linear functions on F, and so they are called *non-linear algorithms*. An example of how $\widetilde{\mathcal{A}}$ induces non-linear functions on F upon fixing a point in R is given in Example 1.

$R = \mathrm{Hom}_{\mathrm{alg}}(A, \mathrm{End}(V)) \times \mathrm{Hom}(F, V)$ is understood as a family of computing machines: a point (w, e) in R fixes how $\mathcal{A}$ acts on V and the framing map $e \in \mathrm{Hom}(F, V)$, and hence entirely determines how an algorithm runs in the machine corresponding to (w, e).

Let us emphasize that the state space V is basis-free. The family of metrics is $\mathrm{GL}(V)$-equivariant: $h_{(\rho,e)}(v,w) = h_{g\cdot(\rho,e)}(g\cdot v, g\cdot w)$ for any $g \in \mathrm{GL}(V)$. Thus, given $a \in \tilde{\mathcal{A}}$, the non-linear functions that a induces for the two machines $r \in R$ and $g \cdot r \in R$ equal to each other. In other words, an algorithm $a \in \tilde{\mathcal{A}}$ drives all machines parametrized by the *moduli stack* $[R/\mathrm{GL}(V)]$ to produce functions on F:

$$\tilde{\mathcal{A}} \times [R/\mathrm{GL}(V)] \to \mathrm{Map}(F,F). \tag{2}$$

As mentioned above, the advantage is that the single near-ring $\tilde{A}$ controls all machines in $[R/\mathrm{GL}(V)]$ and for all V simultaneously (independent of dim V).

In [5], we formulated noncommutative differential forms on a near-ring $\tilde{\mathcal{A}}$, which induce $\mathrm{Map}(F,F)$-valued differential forms on the moduli $[R/\mathrm{GL}(V)]$. It is extended from the Karoubi-de Rham complex [6–9] for algebras to near-rings. (2) above is the special case for 0-forms, which are simply elements in $\tilde{\mathcal{A}}$. The cases of 0-forms and 1-forms are particularly important for gradient descent: recall that gradient of a function is the metric dual of the differential of that function.

Theorem 1 ([5])**.** *There exists a degree-preserving map*

$$DR^\bullet(\tilde{\mathcal{A}}) \to (\Omega^\bullet(R, \mathrm{Map}(F,F)))^{\mathrm{GL}(V)}$$

which commutes with d on the two sides.

A differential form on $\tilde{\mathcal{A}}$ can be recorded as a *form-valued tree*, see the right-hand side of Figure 1. They are rooted trees whose edges are labeled by $\phi \in DR^\bullet(\mathrm{Mat}_F(\hat{\mathcal{A}}))$; leaves are labeled by $\alpha \in \tilde{\mathcal{A}}$; the root is labeled by 1 (if not a leaf); nodes which are neither leaves nor the root are labeled by the symbols $D^{(p)}_{\sigma_\ell}|_\alpha$ that correspond to the p-th order symmetric differentials of σ_ℓ.

In application to machine learning, an algorithm $\tilde{\gamma} \in \widetilde{\mathcal{A}}$ induces a 0-form of $\widetilde{\mathcal{A}}$, for instance

$$\int_K |\tilde{\gamma}(x) - f(x)|^2 dx \tag{3}$$

for a given dataset encoded as a function $f : K \to \mathbb{R}$. This 0-form and its differential induces the cost function and its differential on $[R/G]$, respectively, which are the central objects in machine learning.

The differential forms are G-equivariant by construction. There have been a lot of recent works in learning for input data set that has Lie group symmetry [10–16]. On the other hand, our work has focused on the internal symmetry of the computing machine.

In general, the existence of fine moduli is a big problem in mathematics: the moduli stack $[R/G]$ may be singular and pose difficulties in applying gradient descent. Fortunately, if A is a quiver algebra, its moduli space of framed quiver representations $[R/G]$ is a smooth manifold $\mathcal{M}$ (with respect to a chosen stability condition) [17]. This leads us to deep learning explained in the next section.

3. Deep Learning over the Moduli Space of Quiver Representations

An artificial neural network (see Figure 2 for a simple example) consists of:

1. a *graph* $Q = (Q_0, Q_1)$, where Q_0 is a (finite) set of vertices (neurons) and Q_1 is a (finite) set of arrows starting and ending in vertices in Q_0 (transmission between neurons);
2. a *quiver representation* of Q, which associates a vector space V_i to each vertex i and a linear map w_a (called weights) to each arrow a. We denote by $t(a)$ and $h(a)$ the tail and head of an arrow a, respectively.
3. a non-linear function $V_i \to V_i$ for each vertex i (called an *activation function* for the neuron).

Activation functions are an important ingredient for neural network; on the other hand, it rarely appears in quiver theory. Its presence allows the neural network to produce non-linear functions.

Remark 1. *In the recent past there has been rising interest in the relations between machine learning and quiver representations [5,18–20]. Here, we simply put quiver representation as a part of the formulation of an artificial neural network.*

In many applications, the dimension vector $\vec{d} \in \mathbb{Z}_{\geq 0}^{Q_0}$ is set to be $(1, \ldots, 1)$, i.e., all vector spaces V_i associated with the vertices are one-dimensional. For us, it is an unnecessary constraint, and we allow $\vec{d}$ to be any fixed integer vector.

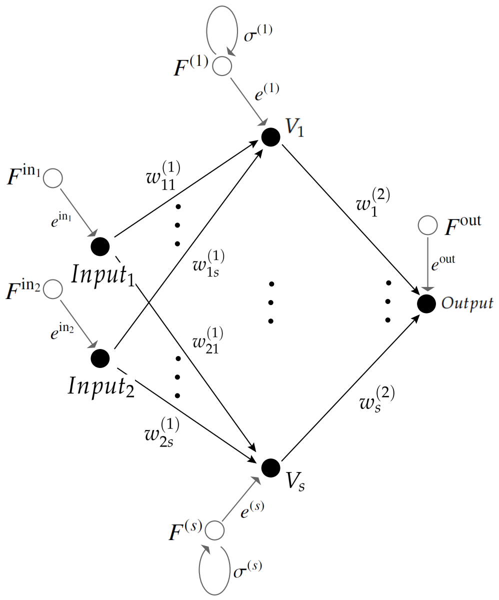

Figure 2. A simple neural network with one hidden layer with s-many neurons. Neural networks in applications are typically much more complicated, but in nature are still quiver representations equipped with activation functions.

Any non-trivial non-linear function $V_i \to V_i$ cannot be $\mathrm{GL}(V_i)$-equivariant. However, in quiver theory, V_i is understood as a basis-free vector space and requires $\mathrm{GL}(V_i)$-equivariance. We resolve this conflict between neural network and quiver theory in [19] using framed quiver representations. The key idea is to put the non-linear function on the framing rather than on the basis-free vector spaces V_i.

Combining with the setting of the last section (Definition 2), we take:

1. $A = \mathbb{C}Q$, the quiver algebra. Elements are formal linear combinations of paths in Q (including the trivial paths at vertices), and product is given by concatenation of paths.
2. $V = \bigoplus_i V_i$, the direct sum of all vector spaces over vertices.
3. Each vertex is associated with a framing vector space F_i. Then $F = \bigoplus_i F_i$.
4. Each point $(w, e) \in R = \mathrm{Hom}_{\mathrm{alg}}(\mathbb{C}Q, \mathrm{End}(V)) \times \bigoplus_i \mathrm{Hom}(F_i, V_i)$ is a framed quiver representation. Specifically, $w \in \mathrm{Hom}_{\mathrm{alg}}(\mathbb{C}Q, \mathrm{End}(V))$ associates a matrix w_a to each arrow a of Q; $e^{(i)} \in \mathrm{Hom}(F_i, V_i)$ are the framing linear maps.
5. The group G is taken to be $\prod_i \mathrm{GL}(V_i)$. An element $g = (g_i)_{i \in Q_0}$ acts on R by
$$g \cdot \left((w_a)_{a \in Q_1}, (e^{(i)})_{i \in Q_0} \right) = \left((g_{h(a)} \cdot w_a \cdot g_{t(a)}^{-1})_{a \in Q_1}, (g_i \cdot e^{(i)})_{i \in Q_0} \right).$$
6. We have (possibly non-linear) maps $\sigma_i : F_i \to F_i$ for each vertex. To match the notation of Definition 2, σ_i can be taken as maps $F \to F$ by extension by zero.

By the celebrated result of [17], we have a fine moduli space of framed quiver representations $\mathcal{M} = \mathcal{M}_{n,d} = R \,/\!/\, G$, where n, d are the dimension vectors for the framing $\{F_i\}_{i \in Q_0}$ and representation $\{V_i\}_{i \in Q_0}$, respectively. In particular, we have the universal vector bundles $\mathcal{V}_i$ over $\mathcal{M}$, whose fiber over each framed representation $[w, e] \in \mathcal{M}$ is the representing vector space V_i over the vertex i.

$\mathcal{V}_i$ plays an important role in our computational model, namely a vector $v \in \mathcal{V}_i$ over a point $[w, e] \in \mathcal{M}$ is the state of the i-th neuron in the machine parametrized by $[w, e]$.

Remark 2. *The topology of $\mathcal{M}$ is well understood by [21] as iterated Grassmannian bundles. Framed quiver representations and their doubled counterparts play an important role in geometric representation theory [22,23].*

To fulfill Definition 2 (see Item (2)), we need to equip each $\mathcal{V}_i$ with a bundle metric h_i, so that the adjoint e^* makes sense. In [19], we have found a *bundle metric that is merely written in terms of the algebra $\mathcal{A}$*. It means the formula works for (infinitely many) quiver moduli for all dimension vectors of representations simultaneously.

Theorem 2 ([19])**.** *For a fixed vertex (i), let ρ_i be the row vector whose entries are all the elements of the form $w_\gamma e^{(t(\gamma))} : R_{n,d} \to \mathrm{Hom}(\mathbb{C}^{n_{t(\gamma)}}, \mathbb{C}^{d_i})$ such that $h(\gamma) = i$. Consider*

$$H_i := \rho_i \rho_i^* = \sum_{h(\gamma)=i} \left(w_\gamma e^{(t(\gamma))} \right) \left(w_\gamma e^{(t(\gamma))} \right)^* \tag{4}$$

as a map $\rho_i \rho_i^* : R_{n,d} \to End(\mathbb{C}^{d_i})$. Then $(\rho_i \rho_i^*)^{-1}$ is $\mathrm{GL}(d)$-equivariant and descends to a Hermitian metric on $\mathcal{V}_i$ over $\mathcal{M}$.

Example 1. *Consider the network in Figure 2. The quiver has the arrows $a^{(1)}_{1,k}, a^{(1)}_{2,k}$ for $k = 1, \ldots, s$ (between the input and hidden layers) and $a^{(2)}_k$ (between the hidden and output layers). In application, we consider the algorithm*

$$\tilde{\gamma} := \sum_{k=1}^{s} \hat{a}^{(2)}_k \sigma_k \circ \sum_{j=1}^{2} \hat{a}^{(1)}_{jk} \in \tilde{\mathcal{A}}$$

where $\hat{a}^{(1)}_{jk} := \left(e^{(j)}\right)^* a^{(1)}_{jk} e^{in_j}$ and $\hat{a}^{(2)}_k := \left(e^{out}\right)^* a^{(2)}_k e^{(k)}$. Please note that the adjoints $\left(e^{(j)}\right)^*$ and $\left(e^{out}\right)^*$ are with respect to the metric H_i and H_{out}, respectively.

$\tilde{\gamma}$ is recorded by the activation tree on the left-hand side of Figure 1, for the case $s = 2$. $\tilde{\gamma}$ drives any activation module (with this given quiver algebra) to produce a function $F_{\mathrm{in}_1} \times F_{\mathrm{in}_2} \to F_{\mathrm{out}}$. For instance, setting the representing dimension to be 1 and taking σ_i to be the ReLu function $\max\{x, 0\}$ on $\mathbb{R}$ is a popular choice. Data passes from the leaves to the root, which is called forward propagation.

Figure 1 shows the differential of $\tilde{\gamma} \in \tilde{\mathcal{A}}$. This 1-form is given by

$$d(a_1\sigma_1 \circ (a_{11} + a_{12}) + a_2\sigma_2 \circ (a_{21} + a_{22}))$$

$$= da_1(\alpha_1 + D^{(1)}_{\sigma_1}|_{\alpha_1}(da_{11} + da_{12})) + da_2(\alpha_2 + D^{(1)}_{\sigma_2}|_{\alpha_2}(da_{21} + da_{22}))$$

where $\alpha_j = \sigma_j \circ (a_{j1} + a_{j2})$. The terms are obtained by starting at the output node and moving backwards through the activation tree, which is well known as the backpropagation algorithm. Please note that this works on the algebraic level and is not specific to any representation.

$d\tilde{\gamma}$ induces a $Map(F_{\mathrm{in}_1} \times F_{\mathrm{in}_2}, F_{\mathrm{out}})$-valued 1-form on $\mathcal{M}_{(n,d)}$. We can also easily produce $\mathbb{R}$-valued 1-forms, for instance by (3).

For stochastic gradient descent over the moduli $\mathcal{M}_{n,d}$, to find the optimal machine, we still need one more ingredient: a metric on $\mathcal{M}_{n,d}$, to turn a one-form to a vector field. Very nicely, the Ricci curvature of the metric (4) given above gives a well-defined metric on $\mathcal{M}_{n,d}$. So far, all the ingredients involved (namely the algorithm $\tilde{\gamma}$, its differential, the bundle metric H_i and the metric on moduli) are purely written in algebraic symbols and work for moduli spaces in all dimensions (n,d) simultaneously.

Theorem 3 ([19])**.** *Suppose Q has no oriented cycles. Then*

$$H_T := \sum_i \partial\bar{\partial} \log det\, H_i = \sum_i (tr(\partial\rho_i)^* H_i \partial\rho_i - tr(H_i\rho_i(\partial\rho_i)^* H_i(\partial\rho_i)\rho_i^*)) \tag{5}$$

defines a Kähler metric on $\mathcal{M}_{n,d}$ for any (n,d).

Example 2. *Let us consider the network of Figure 2 again, with $s = 2$ for simplicity. Let $n = d = (1,1,1,1)$. Over the chart where $e^{(i)} \neq 0$ for all $i = in_1, in_2, 1, 2, out$, the $\mathrm{GL}(d)$-equivariance allows us to assume that $e^{(i)}(e^{(i)})^* = 1$. Then H_{in_j} are trivial for $j = 1,2$, and so $\partial\bar{\partial}\log H_{in_j} = 0$. Let $x_1 = (w^{(1)}_{11}, w^{(1)}_{21})$ and $x_2 = (w^{(1)}_{12}, w^{(1)}_{22})$. We have*

$$\partial\bar{\partial}\log H_j = \partial\bar{\partial}\log\left(1+|x_j|^2\right)^{-1} = \frac{(1+|x_j|^2)dx_j \wedge d\bar{x}^t_j + \bar{x}_j dx^t_j d\bar{x}_j x^t_j}{(1+|x_j|^2)^2};$$

$$\partial\bar{\partial}\log H_{out} = \partial\bar{\partial}\log\left(1+|w^{(2)}_1|^2|x_1|^2+|w^{(2)}_2|^2|x_2|^2\right)^{-1} = \frac{dx_j \wedge d\bar{x}_j + dw^{(2)}_j \wedge d\bar{w}_j{}^{(2)}}{(1+|w^{(2)}_1|^2|x_1|^2+|w^{(2)}_2|^2|x_2|^2)}$$
$$+\frac{(|w^{(2)}_j|^2\bar{x}_j dx^t_j d\bar{x}_j x^t_j) + (|x_j|^2 dw_j \wedge d\bar{w}_j) + (\bar{w}_j x_j dw_j \wedge d\bar{x}_j) + (w_j\bar{x}_j dx_j \wedge d\bar{w}_j)}{(1+|w^{(2)}_1|^2|x_1|^2+|w^{(2)}_2|^2|x_2|^2)^2}.$$

4. Uniformization of Metrics over the Moduli

The original formulation of deep learning is over the flat vector space of representations $\mathrm{Hom}_{\mathrm{alg}}(\mathbb{C}Q, \mathrm{End}(V))$, rather than the moduli space $\mathcal{M} = R\,/\!/\,G$ of framed representations which has a semi-positive metric H_T. In [5], we found the following way of connecting our new approach with the original approach by varying the bundle metric H_i in (4).

We shall assume $n_i \geq d_i\ \forall i$. Let us write the framing map (which is a rectangular matrix) as $e^{(i)} = (\epsilon_i\ b_i)$ where ϵ_i is the largest square matrix and b_i is the remaining part. (In applications b_i usually consists of 'bias vectors'.) This allows us to rewrite Equation (4) in the following way:

$$H_i(\alpha) = \left(\epsilon_i\epsilon_i^* + \alpha_{e^{(i)}} b_i b_i^* + \sum_{\gamma:h(\gamma)=i,\gamma\neq e^{(i)}} \alpha_\gamma w_\gamma e^{t(\gamma)}\left(w_\gamma e^{t(\gamma)}\right)^*\right)^{-1}.$$

with $\alpha = (\alpha_\gamma)_{\gamma:h(\gamma)=i} = (1,\ldots,1)$.

If instead, we set α_γ to different values, then $H_i(\alpha)$ will still be G-equivariant, but it may no longer positive definite on the whole space $\mathcal{M}$. Motivated by the brilliant construction of dual Hermitian symmetric spaces, we define

$$\mathcal{M}(\alpha) := \{[w,e] \in \mathcal{M} : H_i(\alpha) \text{ is positive definite at } [w,e]\}.$$

Elements $[w,e] \in \mathcal{M}(\alpha)$ are called space-like representations with respect to $H_i(\alpha)$. If we set $\alpha = \vec{0}$, it becomes

$$H^0_i := (\epsilon_i\epsilon_i^*)^{-1}$$

and $\mathcal{M}^0 := \mathcal{M}(\alpha = \vec{0})$ is exactly the flat vector space $\mathrm{Hom}_{\mathrm{alg}}(\mathbb{C}Q, \mathrm{End}(V))$. This recovers the original Euclidean learning.

On the other hand, setting $\alpha = (-1, \ldots, -1)$, we obtain a semi-negative moduli space $(\mathcal{M}^-, H_T)$ [5], which is a generalization of the hyperbolic spaces or non-compact dual of Grassmannians. This is a very useful setting as there have been several fascinating works done on machine learning performed over hyperbolic spaces, for example [24–27].

Remark 3. *The above construction gives a family of metrics H_i parametrized by α. α_γ can be interpreted as filtering parameters that encode the importance of the paths γ. It is interesting to compare this with the celebrated attention mechanism. We can build models that use these parameters to filter away noise signals during the learning process. We are investigating the applications in transfer learning, especially in situations where we only have a small sample of representative data.*

Author Contributions: Conceptualization, G.J. and S.-C.L.; writing—original draft preparation, G.J.; writing—review and editing, S.-C.L.; initializing and advising the overall ideas and structures, S.-C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We would like to acknowledge Bernd Henschenmacher, whose communications with us about historical developments of quantum physics has partially inspired this project.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jordan, P. *Zur Axiomatik der Quanten-Algebra*; Verlag der Akademie der Wissenschaften und der Literatur in Mainz, in Komm. F. Steiner Verlag: Mainz, Germany, 1950.
2. Segal, I. Postulates for General Quantum Mechanics. *Ann. Math.* **1947**, *48*, 930. [CrossRef]
3. Liebmann, M.; Ruhaak, H.; Henschenmacher, B. Non-Associative Algebras and Quantum Physics—A Historical Perspective. *arXiv* **2019**, arXiv:1909.04027.
4. Moore, C.; Crutchfield, J.P. Quantum automata and quantum grammars. *Theor. Comput. Sci.* **2000**, *237*, 275–306. [CrossRef]
5. Jeffreys, G.; Lau, S.C. Noncommutative Geometry of Computational Models and Uniformization for Framed Quiver Varieties. *arXiv* **2022**, arXiv:2201.05900.
6. Connes, A. Noncommutative differential geometry. *Publ. Math. l'IHÉS* **1985**, *62*, 41–144. [CrossRef]
7. Cuntz, J.; Quillen, D. Algebra extensions and nonsingularity. *J. Am. Math. Soc.* **1995**, *8*, 251–289. [CrossRef]
8. Ginzburg, V. Lectures on Noncommutative Geometry. *arXiv* **2005**, arXiv:0506603.
9. Tacchella, A. An introduction to associative geometry with applications to integrable systems. *J. Geom. Phys.* **2017**, *118*, 202–233. [CrossRef]
10. Barbaresco, F. Souriau-Casimir Lie Groups Thermodynamics and Machine Learning. In *Geometric Structures of Statistical Physics, Information Geometry, and Learning, SPIGL'20, Les Houches, France, 27–31 July 2021*; Springer: Cham, Switzerland, 2021; pp. 53–83. [CrossRef]
11. Cohen, T.; Welling, M. Group Equivariant Convolutional Networks. In Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; Volume 48, pp. 2990–2999.
12. Cohen, T.; Geiger, M.; Weiler, M. A General Theory of Equivariant CNNs on Homogeneous Spaces. *arXiv* **2019**, arXiv:1811.02017.
13. Cohen, T.; Geiger, M.; Koehler, J.; Welling, M. Spherical CNNs. In Proceedings of the ICLR, Vancouver, BC, Canada, 30 April–3 May 2018.
14. Cohen, T.; Weiler, M.; Kicanaoglu, B.; Welling, M. Gauge Equivariant Convolutional Networks and the Icosahedral CNN. In Proceedings of the International Conference on Machine Learning (ICML), Long Beach, CA, USA, 10–15 June 2019.
15. Cheng, M.; Anagiannis, V.; Weiler, M.; de Haan, P.; Cohen, T.; Welling, M. Covariance in Physics and Convolutional Neural Networks. *arXiv* **2019**, arXiv:1906.02481.
16. de Haan, P.; Cohen, T.; Welling, M. Natural Graph Networks. *arXiv* **2020**, arXiv:2007.08349.
17. King, A. Moduli of representations of finite-dimensional algebras. *Q. J. Math. Oxf. Ser.* **1994**, *45*, 515–530.
18. Armenta, M.A.; Jodoin, P.M. The Representation Theory of Neural Networks. *arXiv* **2020**, arXiv:2007.12213.
19. Jeffreys, G.; Lau, S.C. Kähler Geometry of Quiver Varieties and Machine Learning. *arXiv* **2021**, arXiv:2101.11487.
20. Ganev, I.; Walters, R. The QR decomposition for radial neural networks. *arXiv* **2021**, arXiv:2107.02550.
21. Reineke, M. Framed quiver moduli, cohomology, and quantum groups. *J. Algebra* **2008**, *320*, 94–115. [CrossRef]
22. Nakajima, H. Instantons on ALE spaces, quiver varieties, and Kac-Moody algebras. *Duke Math. J.* **1994**, *76*, 365–416. [CrossRef]

23. Nakajima, H. Quiver varieties and finite-dimensional representations of quantum affine algebras. *J. Am. Math. Soc.* **2001**, *14*, 145–238. [CrossRef]
24. Nickel, M.; Kiela, D. Poincaré Embeddings for Learning Hierarchical Representations. In Proceedings of the NIPS, Long Beach, CA, USA, 4–9 December 2017.
25. Ganea, O.E.; Bécigneul, G.; Hofmann, T. Hyperbolic Entailment Cones for Learning Hierarchical Embeddings. *arXiv* **2018**, arXiv:1804.01882.
26. Sala, F.; Sa, C.D.; Gu, A.; Ré, C. Representation Tradeoffs for Hyperbolic Embeddings. *Proc. Mach. Learn. Res.* **2018**, *80*, 4460–4469.
27. Ganea, O.E.; Bécigneul, G.; Hofmann, T. Hyperbolic Neural Networks. *arXiv* **2018**, arXiv:1805.09112.