

# How to make qubits draw

Alexis Toumi\*

*Quantinuum – Quantum Compositional Intelligence*  
17 Beaumont street, OX1 2NA Oxford, UK

October 27, 2023

## Abstract

We introduce a framework for the generation of visual art with quantum computers. Our approach builds on the recent development of quantum natural language processing (QNLP) defined as a structure-preserving translation from grammar to parameterised quantum circuits. Going from one-dimensional text to two-dimensional drawing, we define a formal grammar for pictorial composition where the two basic rules are horizontal and vertical juxtaposition. We then encode each application of a grammatical rule as a piece of quantum circuit, so that composing these pieces together results in a quantum circuit that computes an artistic judgement of the composition. Finally, we train this quantum artistic judgement, i.e. we optimise the parameters of the quantum circuits, via reinforcement learning from human feedback.

Compared to the now mainstream use of classical deep learning for text-to-image generation (e.g. DALL-E, Midjourney, Stable Difusion, etc.) the advantage of our QNLP approach to the generation of visual art is two-fold: expressivity and explainability. First, we show that quantum computers can in theory be used to express artistic judgements beyond what is possible classically. Then, we argue that the compositionality of our approach allows to move away from black-box models and gets us closer to explainable artificial intelligence. Our framework builds upon solid mathematical foundations in terms of category theory and the intuitive concept of string diagram, it also comes with an open source implementation using the DisCoPy library.

## Introduction

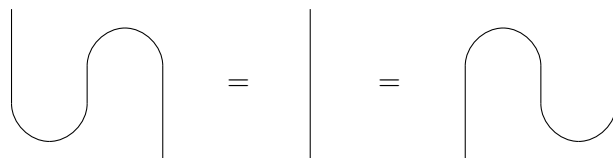
Recent years have seen the explosion of deep learning for text-to-image generation with models such as DALL-E, Midjourney and Stable Difusion demonstrating impressive results and generating a new wave of popular enthusiasm for artificial intelligence (AI). While these models have opened up new possibilities for visual art creation, the black box nature of their billions of parameters trained on billions of examples comes at a price. On the ethical side, they have been criticised for being trained on artworks without the consent of the artists. On the aesthetic side moreover, their opacity limits the ability of artists to incorporate their own preferences into the generated artwork. We argue that artists should seek to open the black box and work with explainable artificial intelligence (XAI) [1].

---

\*[alexis@toumi.email](mailto:alexis@toumi.email)

While today’s deep learning art is all big data and no explainable structure, the work of generative art pioneer Georg Nees was based on computer programs with artfully crafted structure fed with random data [2]. This can be understood as the opposition between two dual principles for AI: distributional and compositional semantics. On one hand, distributional semantics defines the meaning of words from the contexts in which they appear, i.e. their distribution in large datasets. It is the fundamental principle behind deep learning and the connectionist approach to AI. On the other, compositional semantics defines the meaning of sentences as a function of the meaning of words, and of the grammatical rules used to compose them. It is at the basis of the symbolic approach, also called good old-fashioned AI [3].

DisCoCat (Categorical Compositional Distributional) models [4] combine the best of both worlds with the help of category theory, a general theory of mathematical structures and the translations between them. Abstractly, DisCoCat models are monoidal functors from grammar to vector spaces. Concretely, each word is assigned a vector which captures its distributional meaning and the model computes the meaning of sentences by composing these vectors according to their grammatical structure. This form of applied category theory [5] relies heavily on string diagrams to combine mathematical rigour and graphical intuition. The same string diagrams have also been used as a high-level language to formalise quantum mechanics in such an intuitive way that it could be taught in kindergarten [6]. In both quantum mechanics and natural language, string diagrams allow to reason visually about information flow, be it between words in a sentence or gates in a quantum circuit. In fact, the same graphical equation “zig-zag equals straight line equals zag-zig”:



can be used to formalise both grammatical links between words and entanglement between qubits. This grammar-as-entanglement analogy has led to the recent development of quantum natural language processing (QNLP) [7, 8, 9, 10] and to its implementation on quantum hardware [11, 12].

Miranda et al. [13] used this QNLP framework to implement the first proof-of-concept of a quantum algorithm to generate music. In this work, we propose to go from the one-dimensional world of language and music to two-dimensional visual art and show how to make qubits draw. In all three applications, the advantage of QNLP over classical methods is two-fold: explainability and expressivity. With compositionality at their core, our models are explainable “white boxes” with the structure of their input being faithfully encoded in the structure of the quantum circuit that processes it. Moreover, quantum computers can express distributions that are believed to be impossible to simulate with classical means [14]. While this has been used to demonstrate a theoretical quantum advantage for natural language processing [15, 16], the noisy intermediate-scale quantum (NISQ) [17] machines available today are still far from the size required to outperform the billions of parameters of classical deep learning models.

Yet, artists do not have to wait for large-scale fault-tolerant machines to start incorporating quantum computing into their work. Indeed, we demonstrate that it is possible to use a handful of noisy qubits to compose meaningful visual art. In fact, we argue that the constraints of NISQ computers may even allow for a greater freedom of artistic expression than is possible with large

deep learning models. As opposed to the informal practice of prompt engineering,<sup>1</sup> we argue that artists should craft their own formal grammar and use it to teach quantum computers how to generate pieces according to their artistic judgement. As a minimalist example, we define a formal grammar for pictorial composition based on two basic rules: horizontal and vertical juxtaposition. We then translate these two rules in terms of quantum gates so that the resulting circuit evaluates to an artistic judgement of the composition. Finally, we can use this quantum artistic judgement to generate new compositions which will be subject to reinforcement learning from human feedback.

The remainder of this paper will be structured as follows:

- section 1 introduces formal grammars and string diagrams for pictorial composition,
- section 2 introduces the basics of quantum computing and QNLP for artistic judgement,
- section 3 describes the implementation of the framework using the DisCoPy library [18] and discusses current experiments in collaboration with generative artists.

## 1 A formal grammar for pictorial composition

In his lectures on “Painting and the Question of Concepts” [19], Deleuze introduces the notion of diagram as the foundation of his philosophy of painting. He argues that at its core, abstract art is the expression of a binary code: that of horizontal and vertical. Let us attempt to formulate this philosophical insight in terms of formal grammars and string diagrams. To the best of our knowledge, this linguistic approach to visual art goes back to the work of Kirsch [20, 21, 22].

By a formal grammar  $G^2$  we mean a pair of sets  $G_0, G_1$  called symbols and rules with a pair of domain and codomain functions  $\text{dom}, \text{cod} : G_1 \rightarrow G_0^*$  from rules to lists of symbols. We write  $f : x \rightarrow y$  for  $x = \text{dom}(f)$  and  $y = \text{cod}(f)$ . From  $G$ , we define the set of string diagrams as follows:

- each rule  $f : x \rightarrow y$  is also a string diagram, drawn as a box with its domain and codomain as wires coming in from the top and out to the bottom,
- for each list of symbols  $x \in G_0^*$ , the identity  $\text{id}(x) : x \rightarrow x$  is a string diagram, drawn as parallel wires side by side, and in particular the empty diagram  $\text{id}()$  is a string diagram,
- for each pair of string diagrams  $f : x \rightarrow y$  and  $g : y \rightarrow z$ , the composition  $fg : x \rightarrow z$  is a string diagram drawn as the vertical juxtaposition of  $f$  and  $g$  with the codomain of the first connected to the domain of the second,
- for each pair of string diagrams  $f : x \rightarrow y$  and  $f' : x' \rightarrow y'$ , the tensor  $f \otimes f' : xx' \rightarrow yy'$  is a string diagram drawn as the horizontal juxtaposition of  $f$  and  $f'$ .

String diagrams are subject to the following axioms:

- unitality, i.e.  $\text{id}(x)f = f = f\text{id}(y)$  and  $\text{id}() \otimes f = f = f \otimes \text{id}()$ ,
- associativity, i.e.  $f(gh) = (fg)h$  and  $f \otimes (g \otimes h) = (f \otimes g) \otimes h$ ,

<sup>1</sup>For example “prove this theorem in the style of Grothendieck” or “paint this scene in the style of Kandinsky”.

<sup>2</sup>What we call a formal grammar can also be called a semi-Thue system [23] or a monoidal signature [24].

- interchange, i.e.  $\text{id}(x) \otimes \text{id}(y) = \text{id}(xy)$  and  $(fg) \otimes (f'g') = (f \otimes f')(g \otimes g')$ .

A *monoidal category*<sup>3</sup>  $\mathbf{C}$  is a formal grammar where rules and symbols are called objects and arrows, together with functions for identity, composition and tensor which satisfy the axioms above. For instance, **Set** is the category with sets as objects, functions as arrows with the usual notion of identity, composition and the Cartesian product as tensor. For a formal grammar  $G$ , we write  $\mathbf{G}$  for the *free monoidal category* where objects are symbols and arrows are string diagrams.

The axioms of monoidal categories can be understood intuitively as continuous deformations of two-dimensional pictures:

- unitality allows to make wires shorter or longer and to add empty space anywhere,
- associativity allows to stretch string diagrams horizontally and vertically,
- interchange allows to slide boxes up and down, e.g. with  $g = \text{id}(y)$  and  $f' = \text{id}(x')$  we get:

$$\begin{array}{c} x \\ | \\ \boxed{f} \\ | \\ y \end{array} \quad \begin{array}{c} x' \\ | \\ \boxed{g'} \\ | \\ y' \end{array} = \begin{array}{c} x \\ | \\ \boxed{f} \\ | \\ y \end{array} \quad \begin{array}{c} x' \\ | \\ \boxed{g'} \\ | \\ y' \end{array} = \begin{array}{c} x \\ | \\ \boxed{f} \\ | \\ y \end{array} \quad \begin{array}{c} x' \\ | \\ \boxed{g'} \\ | \\ y' \end{array}$$

In fact, the correspondence between the axioms for strict monoidal categories and deformations of two-dimensional pictures goes both way: two string diagrams are equal if and only if they can be deformed continuously into one another without breaking, intersecting or reordering the wires [25].

A *monoidal functor*<sup>4</sup>  $F : \mathbf{C} \rightarrow \mathbf{D}$  is a homomorphism between strict monoidal categories, i.e. a pair of functions from objects to objects and arrows to arrows which respect the domain, codomain, identity, composition and tensor. In particular, a monoidal functor  $F : \mathbf{G} \rightarrow \mathbf{Set}$  is defined by a set  $F(x)$  for each symbol  $x \in G_0$  and a function  $F(f) : F(x_1) \times \cdots \times F(x_n) \rightarrow F(y_1) \times \cdots \times F(y_m)$  for each rule  $f : x_1 \dots x_n \rightarrow y_1 \dots y_m$ . It can then be applied to any string diagram by interpreting wires as identity functions, vertical and horizontal juxtaposition as composition and Cartesian product.

By a vocabulary we mean a designated subset  $V \subseteq G_0$  of symbols called words, by a language we mean a set of lists of words  $L \subseteq V^*$ . We can now define the language of  $G$  as the set  $L(G) = \{w \in V^* \mid \exists f : w \rightarrow s\}$  of all sentences  $w \in V^*$  with a string diagram  $f : w \rightarrow s$  from the list of words into a designated sentence symbol  $s \in G_0$ . Such languages are called recursively enumerable, they form the level zero of the Chomsky hierarchy which is the most general class of languages that can be generated by a Turing machine or any computer algorithm [26]. Going two levels up in the hierarchy we have context-free grammars, where the rules are restricted to have codomain of length one. In this case, the string diagrams are called syntax trees, with the words as leaves and the sentence symbol as root.

We can now define our formal grammar for pictorial composition. It has only one symbol  $G_0 = \{x\}$  and two rules  $G_1 = \{h : xx \rightarrow x, v : xx \rightarrow x\}$  for horizontal and vertical juxtaposition.<sup>5</sup>

<sup>3</sup>We consider only *strict* monoidal categories, i.e. associativity and unit hold up to equality rather than isomorphism.

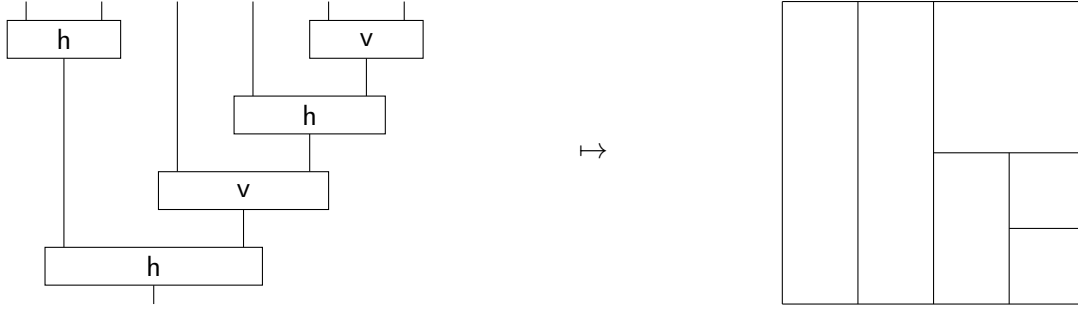
<sup>4</sup>We consider only *strict* monoidal functors, i.e. which respect the tensor up to equality rather than isomorphism.

<sup>5</sup>Note that here the language is trivial, i.e. every list of symbols  $x \dots x$  is a grammatical sentence. In fact, we do not care too much about sentences but rather about their grammatical structure, i.e. about string diagrams. Traditionally, one may add  $h$  and  $v$  as symbols and write the rules as  $x \rightarrow (xhx)$  and  $x \rightarrow (xvx)$  instead.

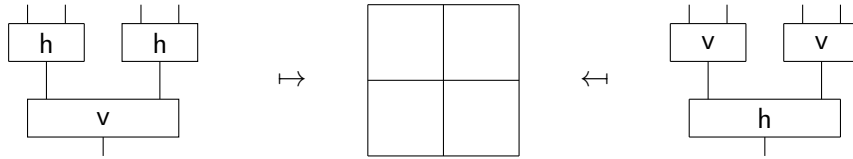
We formalise this interpretation as the monoidal functor  $F : \mathbf{G} \rightarrow \mathbf{Set}$  with  $F(x) = \mathbb{P}([0, 1] \times [0, 1])$  the set of all black-and-white pictures in the unit square together with the following functions:

$$\begin{aligned} F(h) : F(x) \times F(x) &\rightarrow F(x) &:: (a, b) &\mapsto \{(i/2, j)\}_{(i,j) \in a} \cup \{((i+1)/2, j)\}_{(i,j) \in b} \\ F(v) : F(x) \times F(x) &\rightarrow F(x) &:: (a, b) &\mapsto \{(i, j/2)\}_{(i,j) \in a} \cup \{(i, (j+1)/2)\}_{(i,j) \in b} \end{aligned}$$

We define a *pictorial composition* as the interpretation  $F(f)$  of a string diagram  $f : x^{\otimes n} \rightarrow x$  in  $\mathbf{G}$ . It can be visualised as a partition of the unit square into  $n$  rectangles, for example:



Note that two different string diagrams  $f, f' : x^{\otimes n} \rightarrow x$  may have the same interpretation  $F(f) = F(f')$ . For instance,  $f = (h \otimes h)v$  and  $f' = (v \otimes v)h$  both yield the same  $2 \times 2$  grid:



This is in fact another instance of the interchange axiom with  $h$  and  $v$  in place of composition and tensor. Abstractly, we can characterise pictorial compositions as the free magma in the category of magmas, i.e. as the set of  $(h, v)$ -labeled binary trees quotiented by this interchange axiom.

## 2 Quantum-enhanced artistic judgement

## 3 Implementation and experiments

## References

- [1] Amina Adadi and Mohammed Berrada. “Peeking inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)”. In: *IEEE Access* 6 (2018), pp. 52138–52160. DOI: [10.1109/ACCESS.2018.2870052](https://doi.org/10.1109/ACCESS.2018.2870052).
- [2] Margaret A. Boden and Ernest A. Edmonds. “What Is Generative Art?” In: *Digital Creativity* 20.1-2 (June 2009), pp. 21–46. DOI: [10.1080/14626260902867915](https://doi.org/10.1080/14626260902867915).
- [3] John Haugeland. *Artificial Intelligence: The Very Idea*. MIT press, 1989.

- [4] Stephen Clark, Bob Coecke, and Mehrnoosh Sadrzadeh. “Mathematical Foundations for a Compositional Distributional Model of Meaning”. In: *A Festschrift for Jim Lambek*. Ed. by J. van Benthem and M. Moortgat. Vol. 36. Linguistic Analysis. 2010, pp. 345–384. arXiv: [1003.4394](#).
- [5] Tai-Danae Bradley. *What Is Applied Category Theory?* Oct. 2018. DOI: [10.48550/arXiv.1809.05923](#). arXiv: [1809.05923 \[math\]](#).
- [6] Bob Coecke. “Kindergarten Quantum Mechanics”. In: *arXiv:quant-ph/0510032* (Oct. 2005). arXiv: [quant-ph/0510032](#).
- [7] Konstantinos Meichanetzidis et al. “Quantum Natural Language Processing on Near-Term Quantum Computers”. In: *Proceedings 17th International Conference on Quantum Physics and Logic, QPL 2020, Paris, France, June 2 - 6, 2020*. Ed. by Benoît Valiron, Shane Mansfield, Pablo Arrighi, and Prakash Panangaden. Vol. 340. EPTCS. 2020, pp. 213–229. DOI: [10.4204/EPTCS.340.11](#). arXiv: [2005.04147](#).
- [8] Bob Coecke, Giovanni de Felice, Konstantinos Meichanetzidis, and Alexis Toumi. “Foundations for Near-Term Quantum Natural Language Processing”. In: *ArXiv e-prints* (2020). arXiv: [2012.03755](#).
- [9] Bob Coecke, Giovanni de Felice, Konstantinos Meichanetzidis, and Alexis Toumi. “How to Make Qubits Speak”. In: *Quantum Computing in the Arts and Humanities: An Introduction to Core Concepts, Theory and Applications*. Ed. by Eduardo Reck Miranda. Cham: Springer International Publishing, 2022, pp. 277–297. DOI: [10.1007/978-3-030-95538-0\\_8](#).
- [10] Alexis Toumi. “Category Theory for Quantum Natural Language Processing”. PhD thesis. University of Oxford, 2022. arXiv: [2212.06615](#).
- [11] Robin Lorenz et al. “QNLP in Practice: Running Compositional Models of Meaning on a Quantum Computer”. In: *ArXiv e-prints* (Feb. 2021). arXiv: [2102.12846](#).
- [12] Konstantinos Meichanetzidis, Alexis Toumi, Giovanni de Felice, and Bob Coecke. “Grammar-Aware Sentence Classification on Quantum Computers”. In: *Quantum Machine Intelligence* 5.1 (Feb. 2023), p. 10. DOI: [10.1007/s42484-023-00097-1](#).
- [13] Eduardo Reck Miranda et al. “A Quantum Natural Language Processing Approach to Musical Intelligence”. In: *ArXiv e-prints* (Nov. 2021). arXiv: [2111.06741](#).
- [14] Scott Aaronson. “Read the Fine Print”. In: *Nature Physics* 11.4 (Apr. 2015), pp. 291–293. DOI: [10.1038/nphys3272](#).
- [15] William Zeng and Bob Coecke. “Quantum Algorithms for Compositional Natural Language Processing”. In: *Electronic Proceedings in Theoretical Computer Science* 221 (Aug. 2016), pp. 67–75. DOI: [10.4204/EPTCS.221.8](#). arXiv: [1608.01406](#).
- [16] Nathan Wiebe et al. “Quantum Language Processing”. In: *ArXiv e-prints* (Feb. 2019). arXiv: [1902.05162](#).
- [17] John Preskill. “Quantum Computing in the NISQ Era and Beyond”. In: *Quantum* 2 (Aug. 2018), p. 79. DOI: [10.22331/q-2018-08-06-79](#).

- [18] Giovanni de Felice, Alexis Toumi, and Bob Coecke. “DisCoPy: Monoidal Categories in Python”. In: *Proceedings of the 3rd Annual International Applied Category Theory Conference, ACT*. Vol. 333. EPTCS, 2020. DOI: [10.4204/EPTCS.333.13](https://doi.org/10.4204/EPTCS.333.13).
- [19] Gilles Deleuze. “Painting and the Question of Concepts: Lecture 2, 07 April 1981”. In: *Purdue University Research Repository* (1981). DOI: [10.4231/R7PZ571R](https://doi.org/10.4231/R7PZ571R).
- [20] Russell A. Kirsch. “Computer Interpretation of English Text and Picture Patterns”. In: *IEEE Transactions on Electronic Computers* EC-13.4 (Aug. 1964), pp. 363–376. DOI: [10.1109/PGEC.1964.263816](https://doi.org/10.1109/PGEC.1964.263816).
- [21] J L Kirsch and R A Kirsch. “The Structure of Paintings: Formal Grammar and Design”. In: *Environment and Planning B: Planning and Design* 13.2 (June 1986), pp. 163–176. DOI: [10.1068/b130163](https://doi.org/10.1068/b130163).
- [22] Joan L. Kirsch and Russell A. Kirsch. “The Anatomy of Painting Style: Description with Computer Rules”. In: *Leonardo* 21.4 (1988), pp. 437–444. DOI: [10.2307/1578708](https://doi.org/10.2307/1578708). JSTOR: [1578708](https://www.jstor.org/stable/1578708).
- [23] Axel Thue. “Probleme Über Veränderungen von Zeichenreihen Nach Gegebenen Regeln.” Trans. by James F. Power. In: *Natur. KI* 10 (1914). arXiv: [1308.5858](https://arxiv.org/abs/1308.5858).
- [24] Peter Selinger. “A Survey of Graphical Languages for Monoidal Categories”. In: *New Structures for Physics* (2010), pp. 289–355. DOI: [10.1007/978-3-642-12821-9\\_4](https://doi.org/10.1007/978-3-642-12821-9_4).
- [25] André Joyal and Ross Street. “Planar Diagrams and Tensor Algebra”. In: *Unpublished manuscript, available from Ross Street’s website* (1988).
- [26] Noam Chomsky. “Three Models for the Description of Language”. In: *IRE Transactions on Information Theory* 2.3 (Sept. 1956), pp. 113–124. DOI: [10.1109/TIT.1956.1056813](https://doi.org/10.1109/TIT.1956.1056813).