# Quantum computing machine learning

Antonello Corsi –
Senior Research

The 3rd FIT Europe Seminar : Preserving Privacy and Trust in IoT November 15-19, 2021 - Milano, Italy


ENGINEERING

# OUTLINE

- Classical Bits
- Quantum Bits
- Entanglement
- Vazirani Bernstein Algorithm
- Quantum Machine Learning Variational Circuits
- Wrap up

# Representation of a classical bit



$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

# Matrix multiplication

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} ax + by \\ cx + dy \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}$$

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} ax + by + cz \\ dx + ey + fz \\ gx + hy + iz \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} w & x \\ y & z \end{pmatrix} = \begin{pmatrix} aw + by & ax + bz \\ cw + dy & cx + dz \end{pmatrix}$$

# Operations on classical bits



- In quantum computing we will use only reversible operations.
- All the operator that act on a qubit are their own inverse

| | | | | |
|---|---|---|---|---|
| Identity | $f(x) = x$ | $0 \to 0$, $1 \to 1$ | $\begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}1\\0\end{pmatrix}$ | $\begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}0\\1\end{pmatrix}$ |
| Negation | $f(x) = \neg x$ | $0 \to 1$, $1 \to 0$ | $\begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}0\\1\end{pmatrix}$ | $\begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}1\\0\end{pmatrix}$ |
| Constant-0 | $f(x) = 0$ | $0 \to 0$, $1 \to 0$ | $\begin{pmatrix}1 & 1\\0 & 0\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}1\\0\end{pmatrix}$ | $\begin{pmatrix}1 & 1\\0 & 0\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}1\\0\end{pmatrix}$ |
| Constant-1 | $f(x) = 1$ | $0 \to 1$, $1 \to 1$ | $\begin{pmatrix}0 & 0\\1 & 1\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}0\\1\end{pmatrix}$ | $\begin{pmatrix}0 & 0\\1 & 1\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}0\\1\end{pmatrix}$ |

# Tensor product for multiple cbits



We factor back the tensor product to the input vectors
The product state of n bits is a vector 2^n size

**Tensor product for vectors (qubits)**

$$\begin{pmatrix}\alpha_1\\\alpha_2\end{pmatrix} \otimes \begin{pmatrix}\beta_1\\\beta_2\end{pmatrix} = \begin{pmatrix}\alpha_1\begin{pmatrix}\beta_1\\\beta_2\end{pmatrix}\\\alpha_2\begin{pmatrix}\beta_1\\\beta_2\end{pmatrix}\end{pmatrix} = \begin{pmatrix}\alpha_1\beta_1\\\alpha_1\beta_2\\\alpha_2\beta_1\\\alpha_2\beta_2\end{pmatrix}$$

$$00 \equiv \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \quad 01 \equiv \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}0\\1\\0\\0\end{bmatrix}, \quad 10 \equiv \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, \quad 11 \equiv \begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}0\\0\\0\\1\end{bmatrix}$$

# Qubits



- Surprise! We've actually been using qbits all along!
- The cbit vectors we've been using are just special cases of qbit vectors
- A qbit is represented by $\binom{a}{b}$ where $a$ and $b$ are Complex numbers and $||a||^2 + ||b||^2 = 1$
  - The cbit vectors $\binom{1}{0}$ and $\binom{0}{1}$ fit within this definition
  - Don't worry! For this presentation, we'll only use familiar Real numbers.
- Example qbit values:

$$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \quad \begin{pmatrix} \frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{pmatrix} \quad \begin{pmatrix} -1 \\ 0 \end{pmatrix} \quad \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} \end{pmatrix}$$

Caption

# Superposition of base states

## Quantum bits or qubit states:

**Base states:**

$$0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad 1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

**A superposition of base states:**

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \text{where} \quad (|\alpha|)^2 + (|\beta|)^2 = 1$$

- How can a qbit to have a value which is not 0 or 1? This is called superposition.
- Superposition means the qbit is both 0 and 1 and the same time
- When we **measure** the qbit, it **collapses** to an actual value of 0 or 1
  - We usually do this at the end of a quantum computation to get the result
- If a qbit has value $\binom{a}{b}$ then it collapses to 0 with probability $||a||^2$ and 1 with probability $||b||^2$
  - For example, qbit $\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$ has a $\left\|\frac{1}{\sqrt{2}}\right\|^2 = \frac{1}{2}$ chance of collapsing to 0 or 1 (coin flip)
  - The qbit $\binom{1}{0}$ has a 100% chance of collapsing to 0, and $\binom{0}{1}$ has a 100% chance of collapsing to 1

# Operations on qubits

We operate on qubits with matrices

Matrix operators model the effect of some device which manipulates qbit

$$H|0\rangle = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

$$H|1\rangle = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} \end{pmatrix}$$

Caption

# Multiple qbtis

- If the product state of two qbits cannot be factored, they are said to be **entangled**

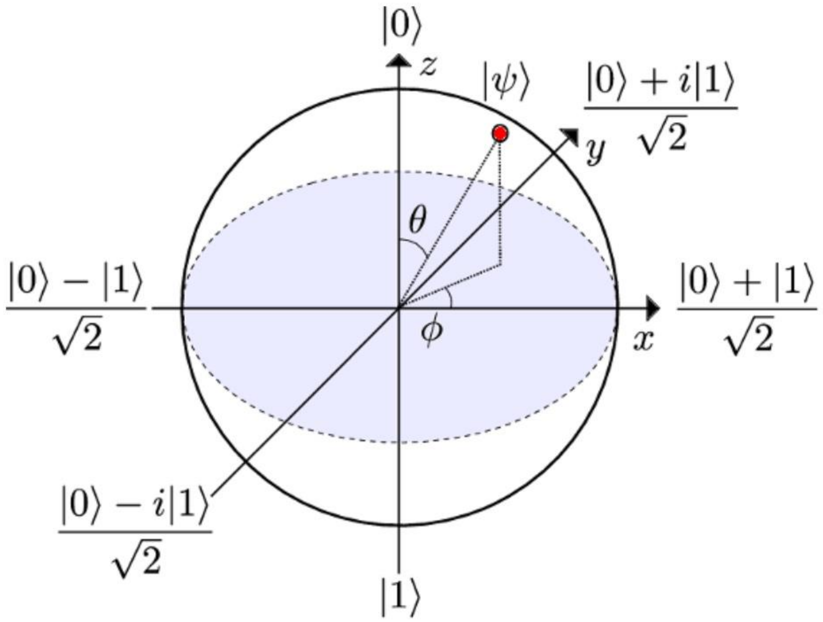
$$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix} \otimes \begin{pmatrix} c \\ d \end{pmatrix}$$

$$ac = \frac{1}{\sqrt{2}}$$
$$ad = 0$$
$$bc = 0$$
$$bd = \frac{1}{\sqrt{2}}$$

- The system of equations has no solution, so we cannot factor the quantum state!
- This has a 50% chance of collapsing to $|00\rangle$ and 50% chance of collapsing to $|11\rangle$

# Bloch Sphere

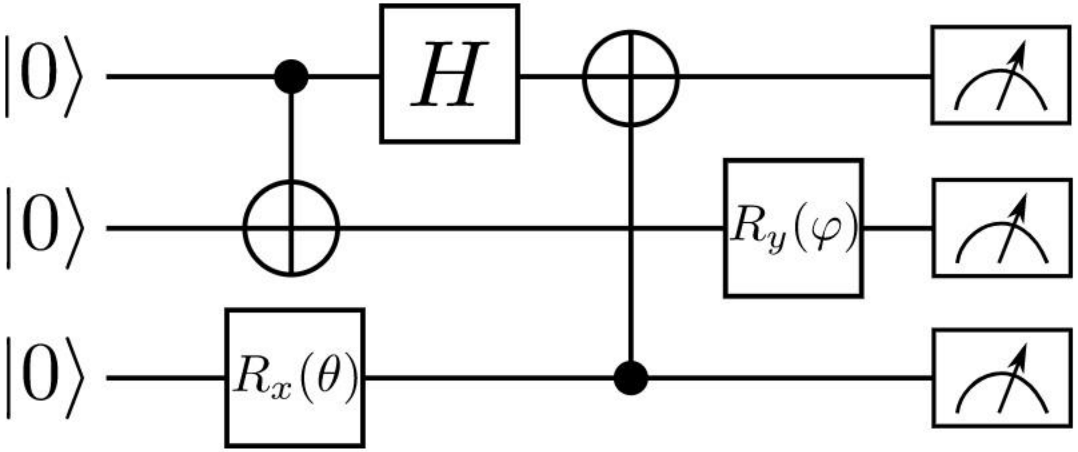
The Bloch sphere is a geometric representation of qubit states as points on the surface of a unit sphere.



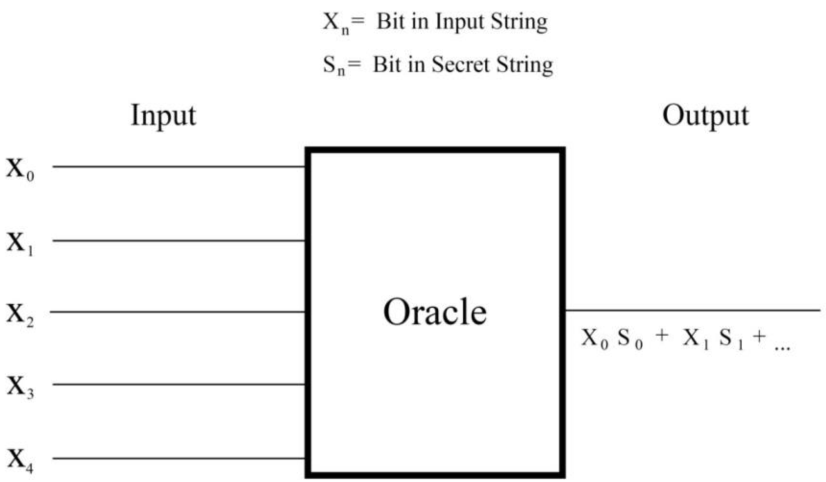

$$|\psi\rangle = e^{i\gamma}\left(\cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle\right)$$

Caption

# Quantum circuit typical architecture

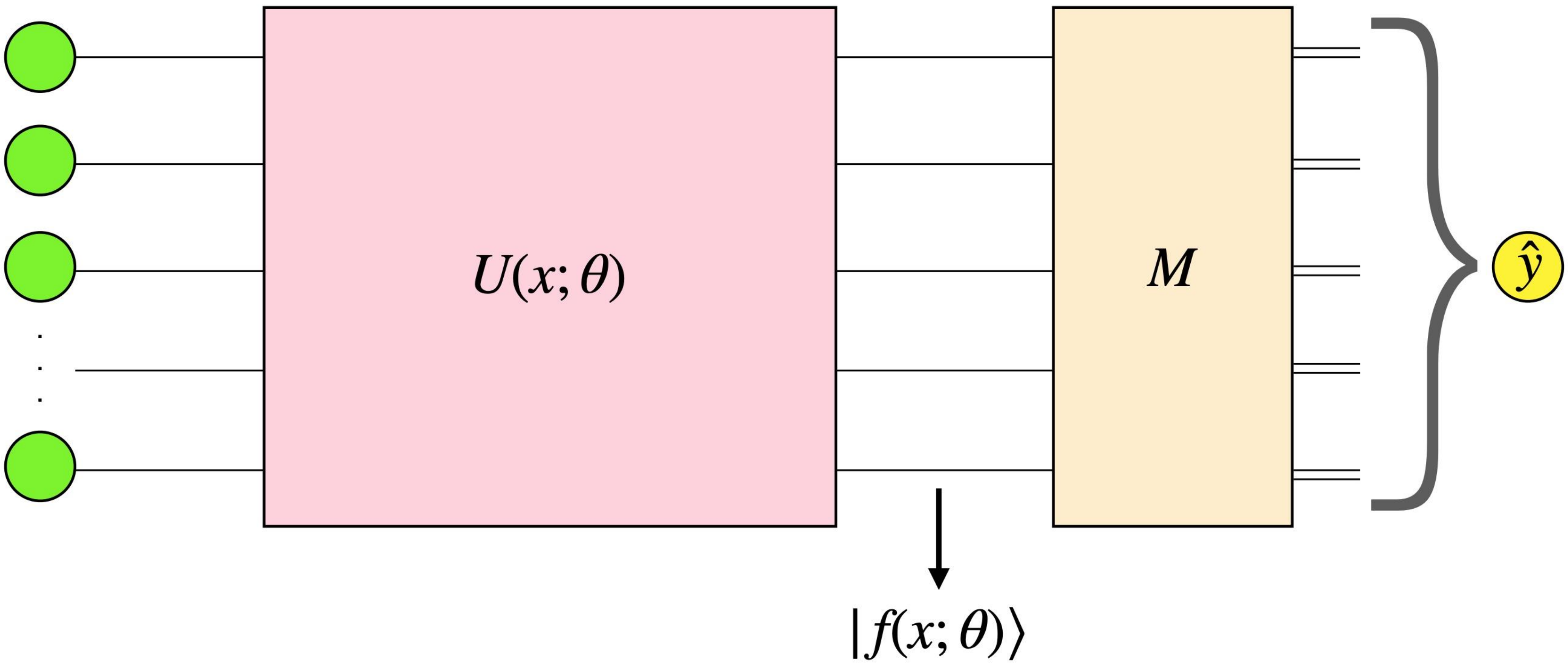


# Vazirani Algorithm for demonstrate QS

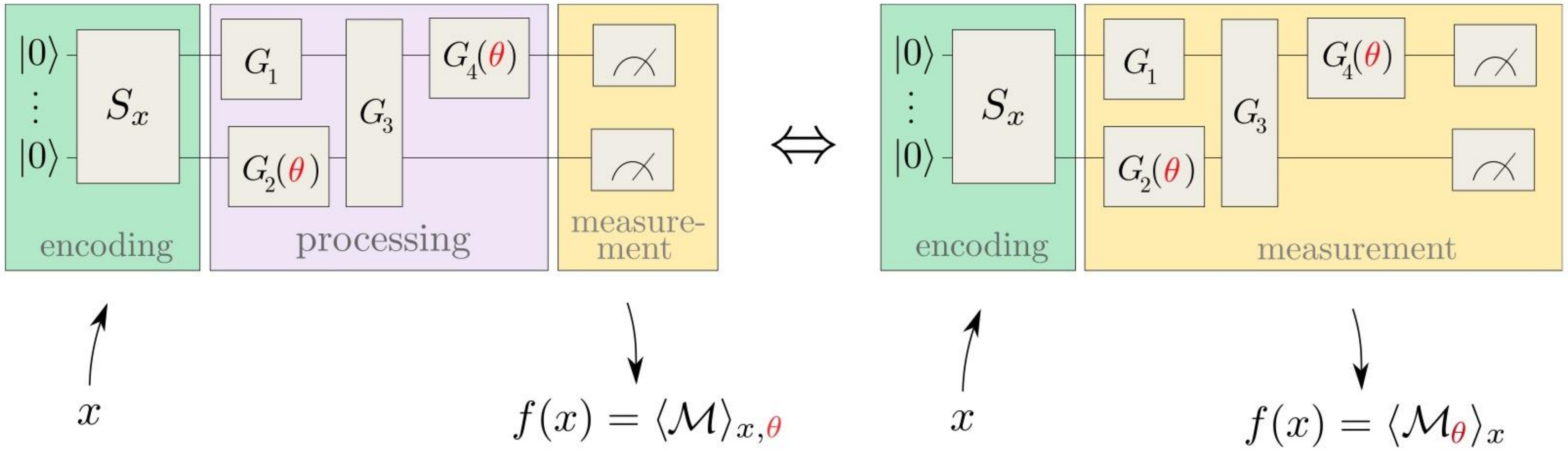



- The Oracle has a secret string of bits, and it is our goal to try to figure it out.
- To do so, we may enter our own string of bits, and in return:
- we get the dot product of the oracles string and ours.

# Quantum Machine Learning



# Variational Circuits



# Wrap Up

In 1995, M.I.T. mathematician Peter Shor, then at AT&T Bell Laboratories, devised a novel algorithm for factoring prime numbers whatever the size.

In 2001, IBM made a quantum computer with seven qubits to demonstrate Shor's algorithm. For qubits, they used atomic nuclei, which have two different spin states that can be controlled through radio frequency pulses.

Quantum states are fragile. It's hard to completely stop qubits from interacting with their outside environment, even with precise lasers in supercooled or vacuum chambers.

Any noise in the system leads to a state called "decoherence," where superposition breaks down and the computer loses information.

# Wrap Up

Error correction It would take a lot of error-correcting qubits—maybe 100 or so per logical qubit--to make the system work. But the end result would be an extremely reliable and generally useful quantum computer.

Other experts are trying to find clever ways to see through the noise generated by different errors. They are trying to build what they call "**Noisy intermediate-scale quantum computers**" using another set of algorithms

Another tactic is to find a new qubit source that isn't as susceptible to noise, such as "topological particles" that are better at retaining information. But some of these exotic particles (or quasi-particles) are purely hypothetical : **Majorana Qubits**

In 2019, Google used a **54-qubit quantum computer named "Sycamore"** to do an incredibly complex (if useless) simulation in under 4 minutes—running a quantum random number generator a million times to sample the likelihood of different results.

# Useful Links

- https://www.youtube.com/watch?v=F_Riqjdh2oMWhen ----→
- https://www.youtube.com/watch?v=Xh9pUu3-WxM
- https://www.youtube.com/watch?v=VPsl_5RQe1A&list=PLnhoxwUZN7-6hB2iWNhLrakuODLaxPTOG
- https://www.youtube.com/channel/UClBNq7mCMf5xm8baE_VMl3A

# Thank you!

**Antonello Corsi**
Senoior Researcher

- www.eng.it
- LifeAtEngineering
- @EngineeringSpa
- Engineering Ingegneria Informatica Spa
- gruppo.engineering