Quantum Hybrid Diffusion Models for Image Synthesis pythorch implementation

Francesca De Falco, Andrea Ceschini1, Alessandro Sebastianelli, Bertrand Le Saux, Massimo Panella.
Implementazione Luca Capece

KI - Künstliche Intelligenz https://doi.org/10.1007/s13218-024-00858-5

Abstract

L'articolo "Quantum Hybrid Diffusion Models for Image Synthesis", propone un'architettura di diffusione ibrida quantistica. Utilizza una U-Net classica con layer ResNet, Self-Attention e rumore gaussiano classico, aggiungendo layer di convoluzione quantistici. Il paper [1] presenta due diversi schemi di ibridazione. Il primo interviene al vertice della rete: i layer convoluzionali ResNet vengono gradualmente sostituiti con circuiti variazionali per creare blocchi Quantum ResNet, quest'ultima è anche il fulcro dell'implementazione. Nella seconda architettura proposta, viene estesa l'ibridazione anche al livello intermedio dell'encoder. Infine le immagini generate dai modelli di diffusione ibridi quantistici sono confrontate con quelle prodotte dai modelli classici e valutate attraverso diverse metriche quantitative. I risultati del paper [1] mostrano un miglioramento nella generazione in tutte le situazioni ma nel caso dell'implementazione in pythorch questo non è sempre vero probabilmente anche dovuto alla variabilità di inizializzazione delle reti.

1. Introduzione

Nel campo dell'AI classica, i modelli di diffusione (Diffusion Models, DMs) si sono affermati come una delle soluzioni più avanzate per la generazione di dati e immagini, superando in termini di qualità e stabilità i Generative Adversarial Networks (GANs). Tuttavia, i DMs possono risultare computazionalmente costosi e richiedere un'accurata ottimizzazione dei parametri.

Il Quantum Machine Learning (QML) è recentemente emerso promettente nell'ambito dell'intelligenza artificiale generativa. Nel contesto quantistico infatti sono stati proposti varianti dei GANs – i Quantum GANs (QGANs) – che hanno dimostrato una maggiore capacità di catturare la distribuzione sottostante dei dati, oltre a richiedere un numero significativamente inferiore di parametri da addestrare. Alcuni di questi approcci sfruttano generatori completamente quantistici, mentre i discriminatori restano classici, mostrando efficacia anche su dataset realistici. Mentre quanto riguarda i modelli di diffusione, sono state esplorate implementazioni di generazione di rumore quantistico da passare nei passaggi di diffusione. In questo lavoro, gli autori propongono per progettare modelli di diffusione ibridi quantistici, combinando circuiti variazionali quantistici con architetture U-Net classiche. Con lo scopo di sfruttare il potere espressivo dei circuiti quantistici in combinazione con la modularità e la flessibilità delle reti neurali tradizionali per migliorarne le prestazioni e ridurne i tempi di convergenza.

2. Background Teorico

2.1. Modelli di Diffusione Classici

Per la teoria dei Diffusion Models classici mi appoggio alla formulazione matematica standard presentata in Denoising Diffusion Probabilistic Models [3]

.

I Diffusion Models si basano su due fasi distinte:

- **Forward process**: Detto processo di diffusione, consiste nell'aggiungere rumore gaussiano classico all'immagine iniziale $x_0 \sim q(x)$ al fine di ottenere un distribuzione imparata $q(x)$ con un aggiunta costante di rumore:

$$x_t = \sqrt{1-\beta_t}\, x_{t-1} + \sqrt{\beta_t}$$

Dove

$$\epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, I), \;\; IID, \quad t = 1, \ldots, T$$

ϵ_t rumore gaussiano
β_t determina la scala della varianza al tempo t

- **Markov Chain**: La catena di Markov si sviluppa come segue per ottenere la distribuzione $q(x_t|x_{t-1})$:

$$q(x_{0:T}) = q(x_0) \prod_{t=1}^{T} q(x_t|x_{t-1})$$

$$q(x_t|x_{t-1}) = \mathcal{N}\left(x_t; \sqrt{1-\beta_t}\, x_{t-1}, \beta_t I\right)$$

L'obiettivo è quindi quello di passare da, $x_0 \sim q(x)$ a $x_T \sim \mathcal{N}(0, I)$:

Con l'utilizzo del "Reparameterization trick" è possibile ottenere il rumore ϵ per ogni t

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

dove $\alpha_t = 1 - \beta_t$ e $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$.

- **Backward process**: Detto processo di denoising, mira a ricostruire l'immagine partendo dalla distribuzione nota $x_T \sim \mathcal{N}(0, I)$ stimando $q(x_{t-1}|x_t)$ (in quanto non ottenibile in forma chiusa):

  sia $p_\theta(x)$ il modello che stima la distribuzione dei dati controllato da un parametro θ si ha quindi:

$$p_\theta(x_T) = \mathcal{N}(0, I)$$

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}\left(\mu_\theta(x_t, t), \Sigma_\theta(x_t, t)\right)$$

  Con le dovute approssimazioni si assume varianza fissa e si stima direttamente il rumore tramite una rete neurale:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\, \epsilon_\theta(x_t, t)\right)$$

  La funzione di Loss viene quindi definita come:

$$\mathcal{L}_t^{\text{simple}} = \mathbb{E}_{x_0 \sim q,\, t,\, \epsilon \sim \mathcal{N}(0,I)} \left[\|\epsilon_\theta(x_t, t) - \epsilon\|^2\right]$$

2.2. Circuiti Quantistici Variazionali (VQCs)

In questo paper [1] viene fatto uso di algoritmi quantistici variazionali (Variational Quantum Algorithms, VQAs). Questi fanno uso di circuiti quantistici parametrizzati chiamati *ansatz*. I circuiti *ansatz* sono composti da porte quantistiche parametrizzate θ che eseguono operazioni sui qbit.

In seguito si riporta in versione semplificata il processo di addestramento di un VQC:

3. Metodologia proposta

In questa sezione vengono descritti i circuiti e le architetture utilizzate nel paper "Quantum Hybrid Diffusion Models for Image Synthesis" [1]. In particolare le architetture proposte sono:

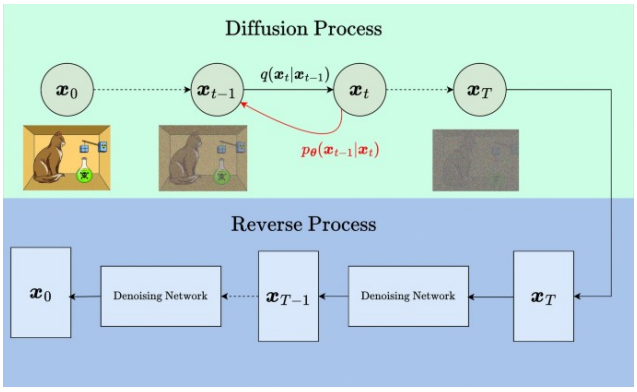


Figure 1. Classical Diffusion Process

Algorithm 1 Addestramento circuito quantistico variazionale (VQC)

1: Codifica dati classici $x \in \mathbb{R}^n$ in spazio di Hilbert $\mathcal{H}^{2^n}$ usando una feature map quantistica $U_\phi(x)$
2: Applicazione ansatz parametrico $U_W(\theta)$ con porte logiche fisse e θ iniziallizzati casualmente allo stato $|\phi(x)\rangle$
3: Completato il circuito si esegue una misurazione rispetto ad una variabile osservabile $\hat{O}$, la predizione è quindi data da:

$$f(x, \theta) = \langle \phi(x)|U_W^\dagger(\theta)\, \hat{O}\, U_W(\theta)|\phi(x)\rangle$$

4: Viene calcolata la funzione di Loss $\mathcal{L}$ e si aggiornano i parametri classicamente con la *parameter-shift rule*

$$\nabla_\theta f(x, \theta) = \frac{1}{2}\left[f\left(x, \theta + \frac{\pi}{2}\right) - f\left(x, \theta - \frac{\pi}{2}\right)\right]$$

- *Quantum Vertex U-Net Hybrid Architecture* dove l'implementazione quantistica avviene al vertice della Unet.

- *Quanvolutional U-Net Hybrid Architecture* dove oltre al vertice viene aggiunta una componente quantistica anche nei layer intermedi, si noti che quest'ultima non è stata implementata per mancanza di risorse computazionali.

Infine si utilizzano metodi di transfert learning per utilizzare l'informazione della Unet classica sulla nuova U-Net ibrida 2.

3.1. Quantum Vertex U-Net Hybrid Architecture

Il modello di partenza è una classica architettura U-Net, a cui il paper [1] propone di aggiungere una componente ibrida in grado di sfruttare le capacità espressive quantistiche

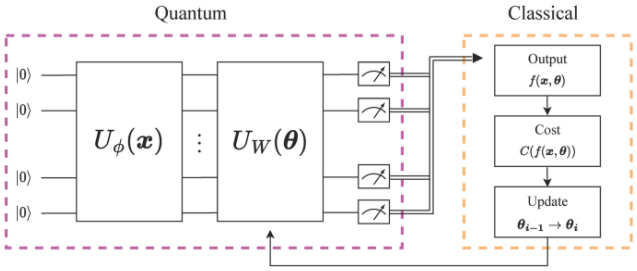


Figure 2. Scheme of a hybrid quantum-classical VQC

per ottenere, a detta dei ricercatori, prestazioni migliori.

L'U-Net classica viene inizialmente implementata secondo la proposta di [3], questa è composta da blocchi ResNet per aggiungere "skip connection" e blocchi Attention per l'aggregazione delle feature; in particolare, nell'implementazione viene utilizzata una Multi-head Attention con quattro teste di attenzione.
I layer ResNet e Attention vengono quindi applicati ai vari livelli di risoluzione nell'U-Net. La prima architettura ibrida proposta, chiamata "Quantum Vertex U-Net" (QVU-Net) 3, utilizza la U-Net come architettura di riferimento e ne integra layer quantistici all'interno della sua struttura.

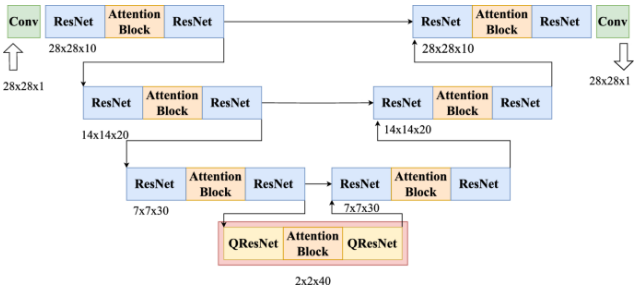


Figure 3. Hybrid U-Net architecture named QVU-Net, where the quantum part is incorporated at the vertex of the network

Per la codifica di dati classici in uno spazio di Hilbert $\mathcal{H}^{2^n}$ viene utilizzato un angle encoding. Questo approccio ha riscontro del singolo pixel sul qbit quindi non necessita di misurazioni di ampiezza, che richiederebbero circuiti esponenzialmente più lunghi e con un numero maggiore di esecuzioni del singolo circuito per generare output statisticamente validi dalla distribuzione degli stati quantistici con la stessa dimensionalità degli input.
Usando l'angle encoding, i dati aventi n input $x \in \mathbb{R}^n$ vengono codificati in n qubit tramite una trasformazione unitaria utilizzando una porta di rotazione Rx; Ogni qubit codifica una feature dei dati di input, ciò consente di avere una profondità del circuito di encoding dei dati pari a O(1).

La U-Net classica elabora inizialmente un'immagine di 28 × 28 × 1 e la scala progressivamente all'interno dell'encoder della rete fino a raggiungere un vertice di dimensioni di 2 × 2 × 40, qui vengono introdotti gli elementi quantistici al vertice della rete. La QResNet è analoga alla ResNet per le componenti di encoder e decoder, la differenza con il blocco ResNet è che alcuni dei livelli convoluzionali utilizzati dal blocco ResNet classico vengono sostituiti con VQC. Poiché lavoriamo con immagini scalate a dimensione 2 × 2, il singolo VQC agirà su 3 canali dell'immagine al posto dei layer convoluzionali, il che equivale di fatto ad un passaggio di filtro sull'intera immagine.

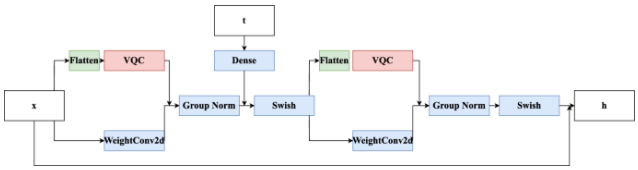


Figure 4. Architecture of the QResNet block, where Convo- lutional layers are replaced with VQCs.

Si noti che non tutti i layer convoluzionali della ResNet vengono sostituiti con VQC; la sostituzione dei layer avviene gradualmente. Nel paper [1] vengono sperimentate vari gradi di ibridazione della rete (10%, 50%, 100%), che per questa architettura specifica corrispondono a quanti canali dello spazio latente 40x2x2 vengono processati dal circuito quantistico. 10% 3 canali 1 circuito 12 qubit, 50% 21 canali 7 circuito 12 qubit, 100% 13 circuti 12 qubit + 1 circuito 4 qbit (Basic Entangling Layer , basato su rotazioni Rx).

La scelta dell'ansatz nei VQC è stata effettuata considerando le dimensioni dello spazio latente al vertice. Poiché l'informazione nel vertice è distribuita su più canali anziché confinata a uno solo, viene adottata una configurazione di ansatz in grado di catturare e sfruttare efficacemente le correlazioni tra questi canali. Il risultato di questi circuiti viene poi misurato su ogni qbit utilizzando una porta PauliZ che restituisce valori reali compresi tra [-1,1] interpretabili dai successivi layer della rete.
Partendo da queste considerazioni, vengono utilizzate due distinte strutture di ansatz. Queste operano su tre canali simultaneamente, con l'obiettivo di elaborare non solo le informazioni locali relative a un singolo canale, ma soprattutto le informazioni intra-canale.

Vengono utilizzati Hierarchical Quantum Convolutional Ansatz (HQConv) 5 i quali estraggono le informazioni locali separatamente tra i canali iniziali (Blocco A), a cui poi seguono ulteriori porte controllate utilizzate per codificare

le informazioni intra-canale (Blocco B). Questi blocchi si possono esprimere matematicamente come:

- Blocco A

$$\left|q_p^2, q_{p+s}^2\right\rangle = [\mathrm{CR}_x(\theta_{x,p}) \circ \mathrm{CR}_z(\theta_{z,p})] \left|q_p^1, q_{p+s}^1\right\rangle \tag{1}$$

$\mathrm{CR}_x(\theta_{x,p})$ e $\mathrm{CR}_z(\theta_{z,p})$ rotazioni controllate sull'asse x e z.
q_p^1 control qbit, q_{p+s}^1 targhet qbit.
s stride, distanza tra targhet e control qbit (s = 1).

- Blocco B

$$\left|q_0^3, q_4^3\right\rangle = [\mathrm{CR}_x(\theta_x) \circ \mathrm{CR}_z(\theta_z)] \left|q_0^2, q_4^2\right\rangle \tag{2}$$

$\mathrm{CR}_x(\theta_{x,p})$ e $\mathrm{CR}_z(\theta_{z,p})$ rotazioni controllate sull'asse x e z.
q_0^2 control qbit, q_4^2 targhet qbit.
s stride, distanza tra targhet e control qbit (s = 1).

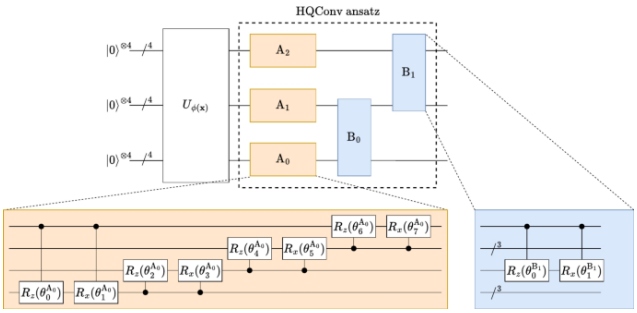


Figure 5. HQConv ansatz

L'altro circuito utilizzato consiste in un Flat Quantum Convolutional Ansatz (FQConv) 6, questo incorpora immediatamente sia le informazioni intra-canale che quelle inter-canale, i blocchi C e D, sono caratterizzati dalla presenza di porte controllate da qubit appartenenti a un altro canale. In termini matematici, i blocchi possono essere espressi come:

-
$$\left|q_p^2, q_{p+s}^2\right\rangle = \mathrm{CR}_z(\theta_{z,p}) \left|q_p^1, q_{p+s}^1\right\rangle \tag{3}$$

-
$$\left|q_p^3, q_{p+s}^3\right\rangle = \mathrm{CR}_x(\theta_{x,p}) \left|q_p^2, q_{p+s}^2\right\rangle \tag{4}$$

Similmente a (HQConv):
q_p^1 control qbit, q_{p+s}^1 targhet qbit, stride (s = 4)

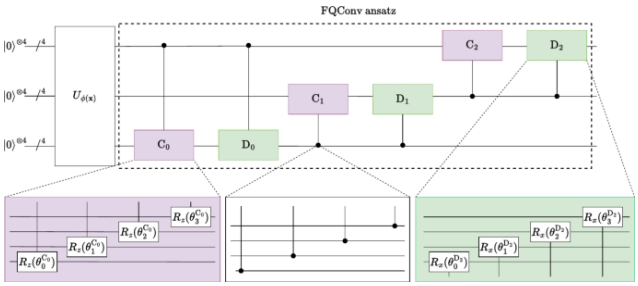


Figure 6. FQConv ansatz

3.2. Transfert Learning

Oltre all'addestramento da zero delle reti ibride e classiche, viene implementato l'utizzo del transfer learning per ridurre i tempi di attesa di addestramento. Questo deriva dal fatto che sia la fase di addestramento che la successiva fase di inferenza, risultano significativamente dispendiose in termini di tempi di addestramento nel caso di QVU-Net ibride, i tempi si dilatano ulteriormente in relazione alla percentuale di circuiti quantistici al vertice. Come illustrato in figura 7, l'idea è quella di addestrare inizialmente una U-Net classica per un certo numero di epoche. I pesi così ottenuti vengono trasferiti alla QVU-Net, ad eccezione di quelli al vertice. Si riaddestra quindi la QVU-Net per un numero limitato di epoche. Al fine di ottenere un miglioramento delle prestazioni rispetto all'utilizzo della sola rete classica, mantenendo tempi di addestramento ridotti.

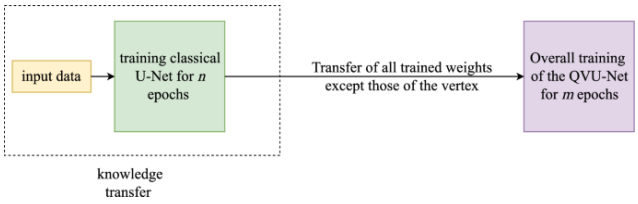


Figure 7. Transfer learning approach

4. Implementazioni

In questa sezione viene descritto il lavoro di implementazione di questa architettura, si noti che nel paper originale [1] vengono proposte ed implementate sia una "Quantum Vertex U-Net" che una "Quanvolutional U-Net Hybrid Architecture" che utilizza circuti quantistici anche a metà dell'encoder/decoder, entrambe con diverse percentuali di ibridazione dei blocchi resNet. A causa dei tempi di esecuzione decisamente onerosi l'implementazione di questo report si limitata a 2 implementazioni:

- *Quantum Vertex U-Net Hybrid Architecture* addestramento completo con un grado di ibridazione del 10% su fashion MNIST.

- *Quantum Vertex U-Net Hybrid Architecture* addestramento usando transfert learning con un grado di ibridazione del 10% su MNIST.

Entrambe le implementazioni sono state testate per entrambi i circuiti quantistici (HQConv, FQConv) descritti nel report. Inoltre nonostante fosse presente del codice relativo al paper [1] questo era scritto interamente in jax + pennylane e non era particolarmente utilizzabile in quanto mancavano componenti, per questo motivo si è optato per reimplementare da zero l'intera architettura utilizzando pytorch + pennylane attenendosi il più possibile al lavoro originale.

4.1. U-net implementation

L'architettura originale della U-net utilizzata del paper [1] non riusciva a mantenere abbastanza informazione al vertice a seguito della compressione dello spazio latente da 30x7x7 a 40x2x2 (Canali,altezza,larghezza), per questo motivo si è optato per un aumento del numero di canali mantenendo la la logica originale delle dimensioni di larghezza ed altezza, in modo da permettere ai circuiti quantistici di continuare ad operare su tre canali.
È stata quindi utilizzata come base l'architettura U-net 8 con il seguente numero di canali:

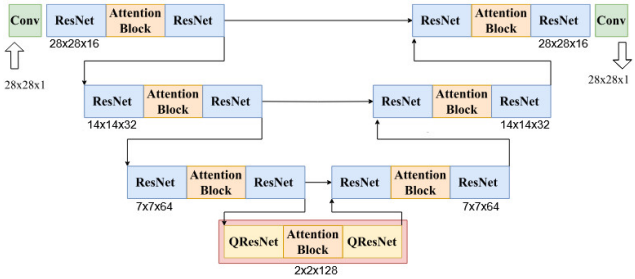


Figure 8. U-net classica

Ne deriva logicamente che il numero di circuiti richiesti per per rendere ibrido al 10% un blocco ResNet aumenta in relazione al numero di canali, in particolare vengono utilizzati (essendo ogni circuito in grado di processare 3 canali) 4 circuiti HQConv/FQConv.

4.2. Implementazione circuiti quantistici

L'implementazione dei circuiti quantistici è stata fatta utilizando pennylane, la struttura dei circuiti hqconv ansatz e fqconv ansatz è la medesima della teoria, entrambi sono contenuti in nei relativi file python, i due circuiti hanno rispettivamente 28 e 24 parametri addestrabili relativi alle loro rispettive porte di rotazione. Inizialmente si è provato ad utilizzare l'implementazione implicità dei layer quantistici di pennylane adattato a pytorch. Questo però non rendeva addestrabili i pesi quindi si è optato per un aggiornamento manuale dei pesi utilizzando *parameter-shift rule* sul quantum node di pennylane.

4.3. Risultati Ottenuti

I modelli sono stati addestrati su due Dataset Fashion MNIST e MNIST, rispettivamente su Fashion MNIST sono stati Addestrati modelli from Scratch mentre per MNIST sono stati addestrati modelli utilizzando transfert learning.
Sono state generate batch da 9 immagini per ogni rete, da queste sono state poi calcolate per ogni modello le metriche utilizzate per verificare la qualità delle immagini, in particolare si è usato FID [2] e KID [5]. Il ridotto numero di immagini è dovuto ai tempi di campionamento estremamente dilatati dovuti anche al fatto che tutti i circuiti quantistici sono vincolati al funzionamento in CPU, con conseguente impossibilità di parallelizzare le varie operazioni di campionamento e addestramento.
Si riportano infine i risultati ottenuti per i modelli sui dataset MNIST e Fashion MNIST:

[hbt!] height**Modello**	**FID**	**KID**
Classical	11.1933	**0.012297 +- 0.000000225**
FQConv	11.5164	0.015437 +- 0.000000014
HQConv	**10.0652**	0.013106 +- 0.000000112

Table 1. MNIST Dataset transfert

In Tabella 1 vengono riportati i risultati dei modelli che utilizzano transfert learning, Il modello di partenza classico è addestrato per un totale di 20 epoche, questo viene poi utilizzato fermando i pesi classici come base per addestrare per ulteriori 5 epoche i circuiti quantistici presenti al vertice della U-net.

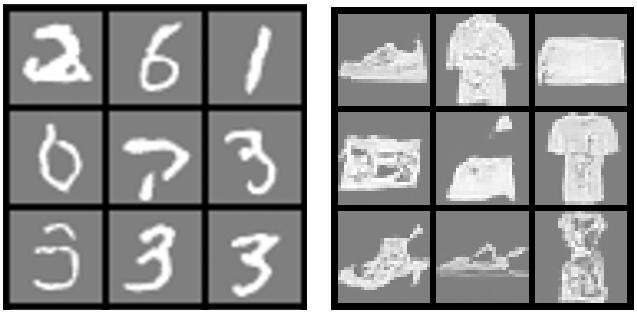
[hbt!] height**Modello**	**FID**	**KID**
Classical	**1.6832**	**0.000800 +- 0.000000225**
FQConv	1.7446	0.000953 +- 0.000000195
HQConv	2.1614	0.001732 +- 0.000000515

Table 2. Fashion MNIST Dataset

In Tabella 2 vengono invece riportati i risultati dei modelli addestrati da zero, con un tasso di ibridazione della rete pari al 10%, tutti i modelli sono addestrati per un totale di 20 epoche.

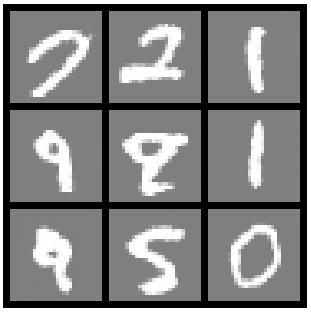
4.4. Conclusioni

Dai risultati ottenuti si riconfermano i risultati ottenuti nel paper [1], un tasso di ibridazione della rete basso produce risultati lievemente peggiori rispetto alla versione classica, nel paper originale infatti si vede come tassi di ibridazione del modello estremi come 10% o 100% producano risultati peggiori, mentre la percentuale di

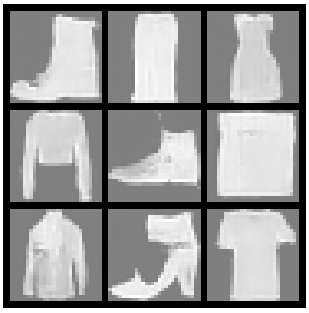


(a) Classical MNIST
FID: 11.19, KID: 0.0123

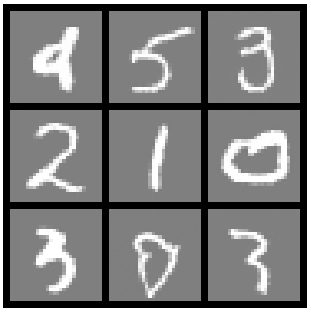
(b) Classical FashionMNIST
FID: 1.68, KID: 0.0008



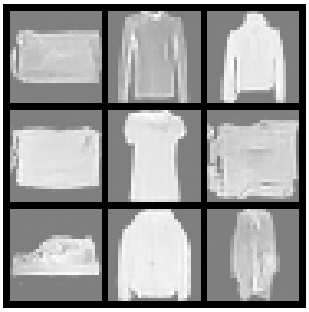
(c) FQConv MNIST
FID: 11.52, KID: 0.0154



(d) FQConv FashionMNIST
FID: 1.74, KID: 0.00095



(e) HQConv MNIST
FID: 10.07, KID: 0.0131



(f) HQConv FashionMNIST
FID: 2.16, KID: 0.00173

Figure 9. Risultati modelli sui dataset MNIST e FashionMNIST.

ibridazione ottimale dimostrano essere intorno al 50%. Si nota infatti come su Fashion MNIST il modello con la migliore performance sia in termini di FID che KID rimanga quello classico seguito dal modello Vertex U-Net con circuito FQconv, mentre la Vertex U-Net con circuito HQConv.

Per quanto riguarda la componente di transfert learnig anche qui si riconfermano i risultati ottenuti nel paper [1], una rete classica con un ulteriore addestramento di una componente quantistica può aggiungere maggiore espressività al modello migliorandone i risultati. Si noti infatti come la Vertex U-Net con circuito HQConv ottenga un FID score maggiore rispetto a gli altri modelli, e nonstante abbia un KID score peggiore rispetto al modello classico si noti come quest'ultimo sia fortemente influenzato dalla dimensione del campione potenzialmente sottostimando la variabilità di quest'indice.
È quindi plausibile che con un ulteriore processo di campionamento dalla stessa rete i risultati ottenuti convergano confermando come migliore la Vertex U-Net con circuito HQconv. Da questa breve analisi si conclude come utilizzare layer quantistici al posto delle convoluzioni classiche possa aumentare l'espressività della rete, ma al momento, almeno con i circuiti testati, una completa sostizione delle convoluzioni non garantisce un assoluto miglioramento delle prestazioni.

Tutto il codice dell'implementazione può essere trovato su github [4].

References

[1] Francesca De Falco, Andrea Ceschini, Alessandro Sebastianelli, Bertrand Le Saux, and Massimo Panella. Quantum hybrid diffusion models for image synthesis. *KI - Künstliche Intelligenz*, 38(4):311–326, Dec 2024. 1, 2, 3, 4, 5, 6

[2] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018. 5

[3] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *CoRR*, abs/2006.11239, 2020. 1, 3

[4] Luca Capece. Quantum-hybrid-diffusion-models, 2025. 6

[5] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. 5