QRKT-GAN: Neural ODE-Inspired Generative Adversarial Network with Numerical Runge-Kutta Methods for Quantum Visual Transformer-Based Generator and Discriminator

Cătălin-Alexandru Rîpanu
Supervisor: Ș.l. dr. ing. Dumitru-Clementin Cercel

Faculty of Automatic Control and Computers,
National University of Science and Technology POLITEHNICA Bucharest

July 2, 2024

Context

In Artificial Intelligence, **Deep Learning** models:

Context

In Artificial Intelligence, **Deep Learning** models:

1. demonstrated **remarkable results** across various domains:

Context

In Artificial Intelligence, **Deep Learning** models:

1. demonstrated **remarkable results** across various domains:
   - Object Classification (Computer Vision)
   - Image Segmentation (Computer Vision)
   - Sentiment Analysis (Natural Language Processing)
   - Synthetic Data Generation

Context

In Artificial Intelligence, **Deep Learning** models:

1. demonstrated **remarkable results** across various domains:
   - Object Classification (Computer Vision)
   - Image Segmentation (Computer Vision)
   - Sentiment Analysis (Natural Language Processing)
   - Synthetic Data Generation
2. greatly improved human lives (Medical Image Recognition [1])

Context

In Artificial Intelligence, **Deep Learning** models:

1. demonstrated **remarkable results** across various domains:
   - Object Classification (Computer Vision)
   - Image Segmentation (Computer Vision)
   - Sentiment Analysis (Natural Language Processing)
   - Synthetic Data Generation
2. greatly improved human lives (Medical Image Recognition [1])

however. . .

Context

In Artificial Intelligence, **Deep Learning** models:

1. demonstrated **remarkable results** across various domains:
   - Object Classification (Computer Vision)
   - Image Segmentation (Computer Vision)
   - Sentiment Analysis (Natural Language Processing)
   - Synthetic Data Generation
2. greatly improved human lives (Medical Image Recognition [1])

however...

for such performance ... millions to billions of neurons are required.

Context (2)

Numerous solutions have been developed to mitigate this problem:

Context (2)

Numerous solutions have been developed to mitigate this problem:

1. Grid Search and Random Search [2, 3]
2. Group Sparsity Regularizers [4]
3. Dimensionality Reduction and Kernel-Sharing [5, 6]
4. Network Weights Splitting [7]
5. Particle Swarm Optimization [8]

Context (2)

Numerous solutions have been developed to mitigate this problem:

1. Grid Search and Random Search [2, 3]
2. Group Sparsity Regularizers [4]
3. Dimensionality Reduction and Kernel-Sharing [5, 6]
4. Network Weights Splitting [7]
5. Particle Swarm Optimization [8]

unfortunately. . .

Context (2)

Numerous solutions have been developed to mitigate this problem:

1. Grid Search and Random Search [2, 3]
2. Group Sparsity Regularizers [4]
3. Dimensionality Reduction and Kernel-Sharing [5, 6]
4. Network Weights Splitting [7]
5. Particle Swarm Optimization [8]

unfortunately. . .

these **classical** methods have inherent drawbacks in their logic.

Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs)

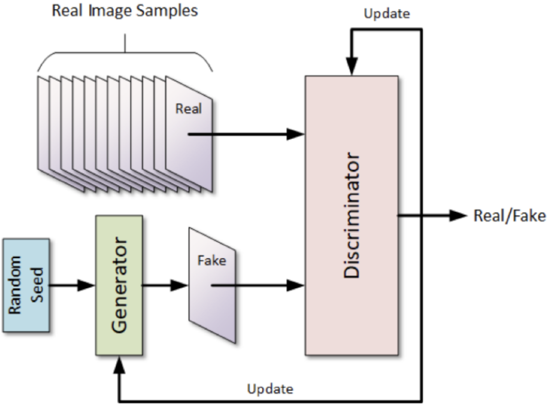

Figure: The Generative Adversarial Network Architecture [9]

The Transformer

The Transformer

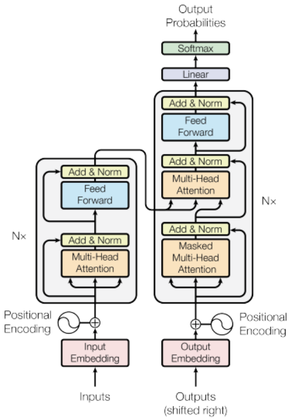

Figure: The Transformer Architecture [10]

The Visual Transformer

The Visual Transformer

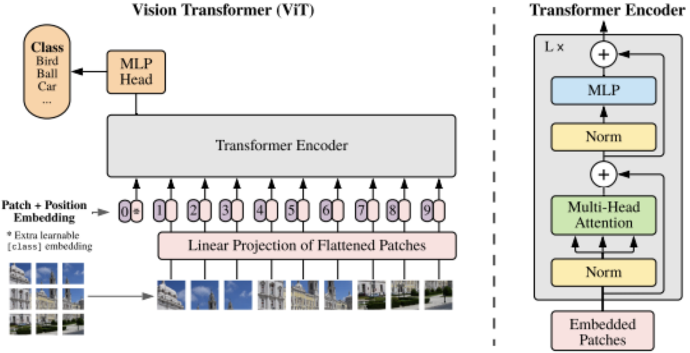

Figure: The Visual Transformer Architecture [10, 11]

The Visual Transformer (2)

Thus, the transformation at each layer is defined as:

The Visual Transformer (2)

Thus, the transformation at each layer is defined as:

$$W = X + \mathrm{MHA}(\mathrm{Norm}(X), \mathrm{Norm}(X), \mathrm{Norm}(X)) \quad (1)$$

$$X' = W + \mathrm{MLP}(\mathrm{Norm}(W)) \quad (2)$$

where $X, X' \in \mathbb{R}^{N \times D}$.

The Visual Transformer (3)

And. . . the Multi-Head Attention Mechanism can be expressed as:

The Visual Transformer (3)

And...the Multi-Head Attention Mechanism can be expressed as:

$$\text{Attention}(V, K, Q) = \text{softmax}\left(\frac{QK^T}{\sqrt{D_k}}\right)V \quad (3)$$

$$\text{MHA}(V, K, Q) = \text{Concat}(\text{single\_head}_i)W^O, i = 1 : h \quad (4)$$

$$\text{single\_head}_i = \text{Attention}(VW_i^V, KW_i^K, QW_i^Q) \quad (5)$$

The Visual Transformer (3)

And. . . the Multi-Head Attention Mechanism can be expressed as:

$$\text{Attention}(V, K, Q) = \text{softmax}\left(\frac{QK^T}{\sqrt{D_k}}\right) V \tag{3}$$

$$\text{MHA}(V, K, Q) = \text{Concat}(\text{single\_head}_i) W^O, i = 1 : h \tag{4}$$

$$\text{single\_head}_i = \text{Attention}(VW_i^V, KW_i^K, QW_i^Q) \tag{5}$$

where:

- $W_i^K \in \mathbb{R}^{D_x \times D_k}$, $W_i^V \in \mathbb{R}^{D_x \times D_v}$, $W_i^Q \in \mathbb{R}^{D_x \times D_k}$
- $W^O \in \mathbb{R}^{hD_v \times D_x}$

The Visual Transformer (4)

Let $Y^m = [y_1^m, y_2^m, \ldots, y_L^m]$ [12]:

The Visual Transformer (4)

Let $Y^m = [y_1^m, y_2^m, \ldots, y_L^m]$ [12]:

$$\hat{y}_i^m = y_i^m + G(y_i^m, Y^m), \quad 1 \leq i \leq L, \tag{6}$$

The Visual Transformer (4)

Let $Y^m = [y_1^m, y_2^m, \dots, y_L^m]$ [12]:

$$\hat{y}_i^m = y_i^m + G(y_i^m, Y^m), \quad 1 \leq i \leq L, \tag{6}$$

The output $\hat{Y}^m = [\hat{y}_1^m, \hat{y}_2^m, \dots, \hat{y}_L^m]$ is then fed to the MLP:

$$y_i^{m+1} = \hat{y}_i^m + H(\hat{y}_i^m), \quad 1 \leq i \leq L, \tag{7}$$

The Visual Transformer (4)

Let $Y^m = [y_1^m, y_2^m, \ldots, y_L^m]$ [12]:

$$\hat{y}_i^m = y_i^m + G(y_i^m, Y^m), \quad 1 \leq i \leq L, \tag{6}$$

The output $\hat{Y}^m = [\hat{y}_1^m, \hat{y}_2^m, \ldots, \hat{y}_L^m]$ is then fed to the MLP:

$$y_i^{m+1} = \hat{y}_i^m + H(\hat{y}_i^m), \quad 1 \leq i \leq L, \tag{7}$$

Over the time interval $[m, m+1]$, using Lie-Trotter decomposing method [13, 14]:

$$\frac{dy_i}{dt} = H(y_i) + G(y_i, Y) \tag{8}$$

Neural Runge-Kutta Method (RK4)

Let $F(y_i, Y) = H(y_i) + G(y_i, Y)$

Neural Runge-Kutta Method (RK4)

Let $F(y_i, Y) = H(y_i) + G(y_i, Y)$

In this context, Runge-Kutta method can be written as:

Neural Runge-Kutta Method (RK4)

Let $F(y_i, Y) = H(y_i) + G(y_i, Y)$

In this context, Runge-Kutta method can be written as:

$$y_i(t+1) = y_i(t) + \sum_{j=1}^{n} \gamma_j F_{ij} \tag{9}$$

$$F(y_i, Y) = F_i \tag{10}$$

$$F_{ij} = F_i(y_i + \sum_{p=1}^{j-1} \beta_{jp} F_{ip}, Y) \tag{11}$$

Neural Runge-Kutta Method (RK4)

Let $F(y_i, Y) = H(y_i) + G(y_i, Y)$

In this context, Runge-Kutta method can be written as:

$$y_i(t+1) = y_i(t) + \sum_{j=1}^{n} \gamma_j F_{ij} \tag{9}$$

$$F(y_i, Y) = F_i \tag{10}$$

$$F_{ij} = F_i(y_i + \sum_{p=1}^{j-1} \beta_{jp} F_{ip}, Y) \tag{11}$$

Thus

Neural Runge-Kutta Method (RK4)

Let $F(y_i, Y) = H(y_i) + G(y_i, Y)$

In this context, Runge-Kutta method can be written as:

$$y_i(t+1) = y_i(t) + \sum_{j=1}^{n} \gamma_j F_{ij} \tag{9}$$

$$F(y_i, Y) = F_i \tag{10}$$

$$F_{ij} = F_i(y_i + \sum_{p=1}^{j-1} \beta_{jp} F_{ip}, Y) \tag{11}$$

Thus

$$y_i(t+1) = y_i(t) + \frac{1}{6}(F_{i1} + 2F_{i2} + 2F_{i3} + F_{i4}) \tag{12}$$

The Quantum Visual Transformer

The Quantum Visual Transformer

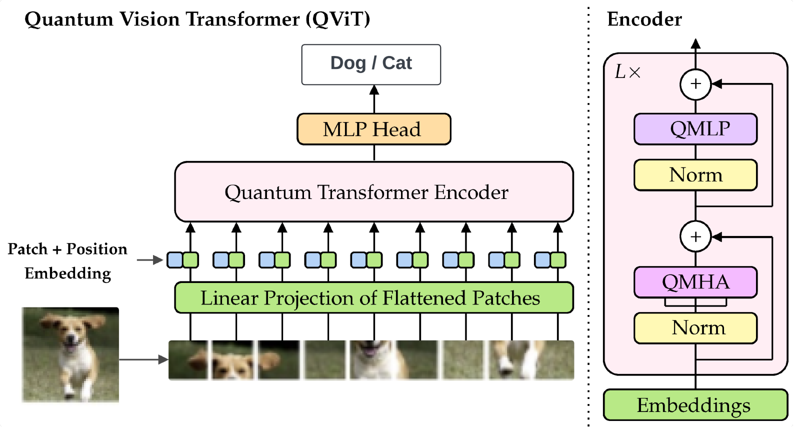

Figure: Quantum Visual Transformer [15]

Quantum Gates

Quantum Gates

$$\mathrm{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$R_X(\theta) = \begin{bmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{bmatrix}$$

$$R_Z(\theta) = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}$$

$$R_Y(\theta) = \begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{bmatrix}$$

The Variational Quantum Circuit

One can use such techniques...

The Variational Quantum Circuit

One can use such techniques. . . only in another reality.

The Variational Quantum Circuit

One can use such techniques... only in another reality.

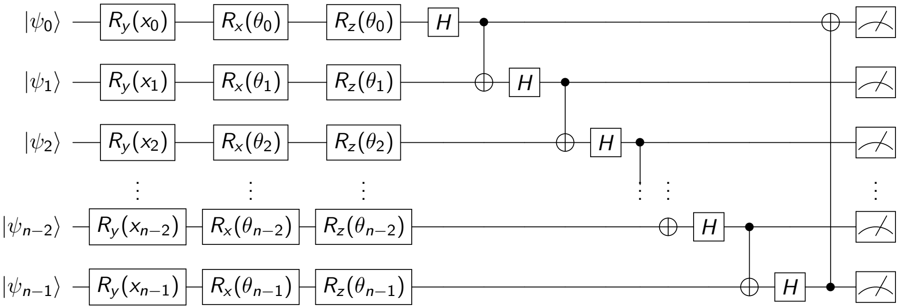

Figure: The Variational Quantum Circuit used in QRKT-GAN

Proposed Solution (QRKT-GAN)

Proposed Solution (QRKT-GAN)

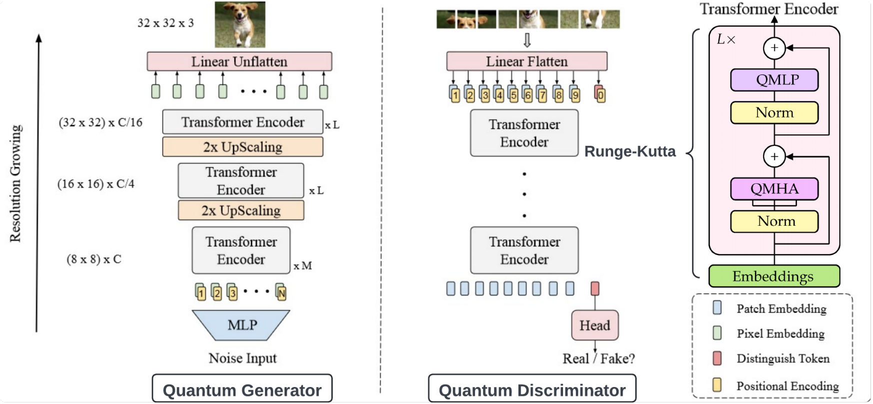

Figure: The QRKT-GAN Architecture. Image inspired from [15, 16]

MNIST Classification

[1] https://www.tensorflow.org/datasets/catalog/mnist

MNIST Classification

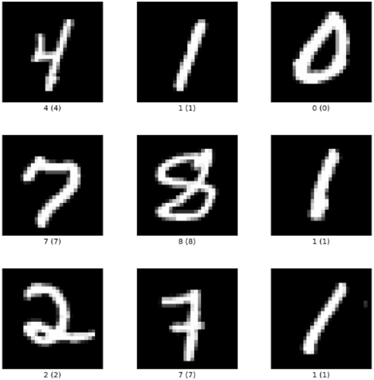

Figure: Examples from MNIST[1]

[1] https://www.tensorflow.org/datasets/catalog/mnist

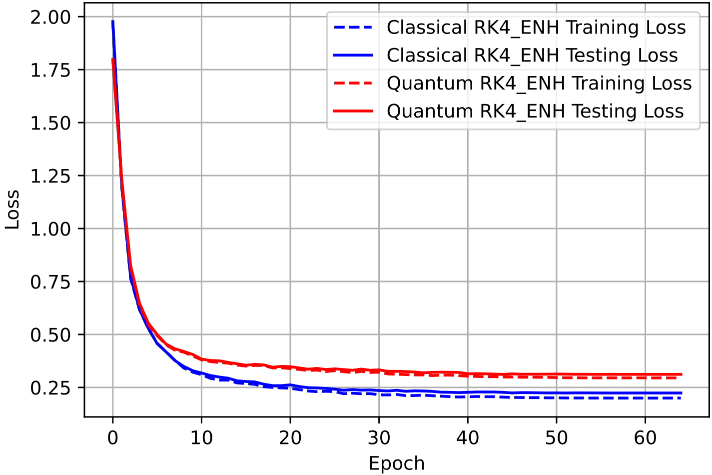

Figure: Cross-entropy loss evolution during learning

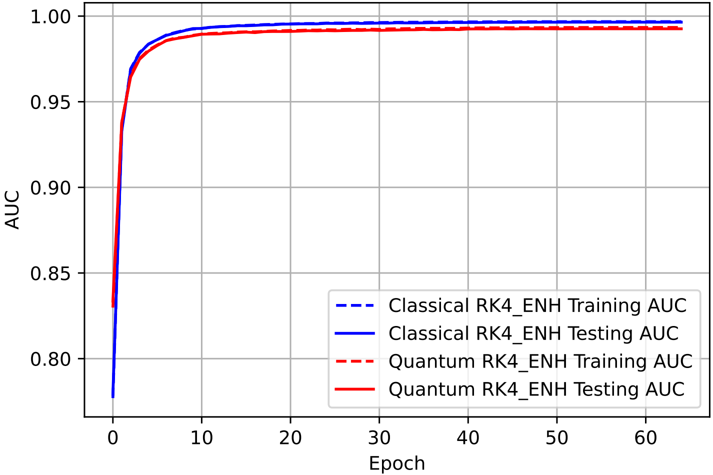

Figure: AUC score evolution during learning

Configurations:

Configurations:

- **Patch Size**: 14
- **Hidden Size**: 6
- **Classical and Quantum ODE-Transformer Blocks**: 3
- **Classical and Quantum Attention Heads**: 2
- **Hidden QMLP Size**: 3

Configurations:

- **Patch Size**: 14
- **Hidden Size**: 6
- **Classical and Quantum ODE-Transformer Blocks**: 3
- **Classical and Quantum Attention Heads**: 2
- **Hidden QMLP Size**: 3

ODE	Train Time (s)	Accuracy	F1 Score	Best AUC Epoch	# Parameters	# Qubits
RK4_ENH	**1842.04**	**95%**	**95%**	54	5971	-
QRK4_ENH	3539.44	91%	91%	**48**	**3520**	357

Table: MNIST metrics for the quantum and classical configurations

CIFAR-10 Classification

[2] `https://www.tensorflow.org/datasets/catalog/cifar10`

CIFAR-10 Classification

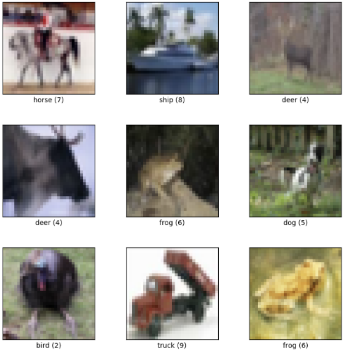

Figure: Examples from CIFAR-10[2]

[2] https://www.tensorflow.org/datasets/catalog/cifar10

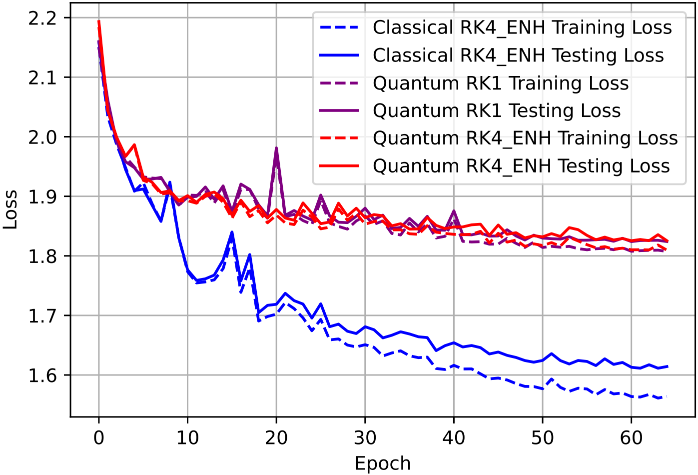

Figure: Cross-entropy loss evolution during learning

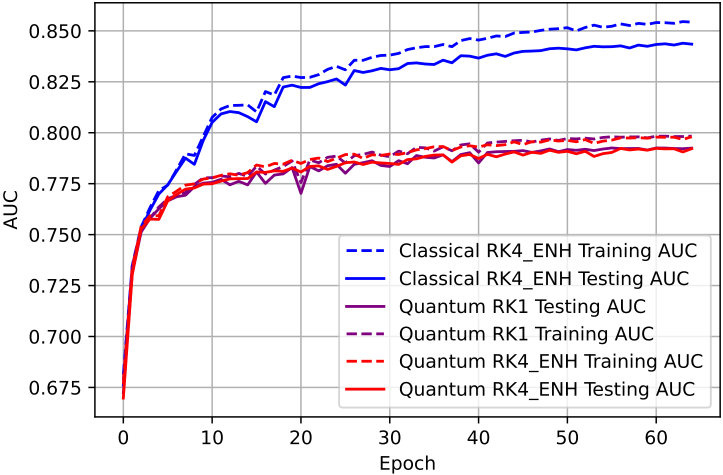

Figure: AUC score evolution during learning

Configurations:

Configurations:

- **Patch Size**: 16
- **Hidden Size**: 12
- **Classical and Quantum ODE-Transformer Blocks**: 1
- **Classical and Quantum Attention Heads**: 6
- **Hidden QMLP Size**: 6

Configurations:

- **Patch Size**: 16
- **Hidden Size**: 12
- **Classical and Quantum ODE-Transformer Blocks**: 1
- **Classical and Quantum Attention Heads**: 6
- **Hidden QMLP Size**: 6

ODE	Train Time (s)	Accuracy	F1 Score	Best AUC Epoch	# Parameters	# Qubits
RK4_ENH	**1685.04**	**42%**	**42%**	65	33634	-
QRK4_ENH	16724.79	34%	33%	61	**20590**	390
QRK1	10909.40	33%	33%	**56**	**20590**	**336**

Table: CIFAR-10 metrics for the quantum and classical configurations

IMDb Classification

[3] https://www.tensorflow.org/datasets/catalog/imdb_reviews

IMDb Classification

Label	Text
0 (neg)	"I have been known to fall asleep during films, but this is usually due to a combination of things including, really tired, being warm and comfortable on the settee and having just eaten a lot. However on this occasion I fell asleep because the film was rubbish [...]"
1 (pos)	"This is a film which should be seen by anybody interested in, effected by, or suffering from an eating disorder. It is an amazingly accurate and sensitive portrayal of bulimia in a teenage girl, its causes and its symptoms. The girl is played by one of the most brilliant young actresses working in cinema today, Alison Lohman, who was later so spectacular in 'Where the Truth Lies' [...]"

Table: Movie Reviews[3]

[3] https://www.tensorflow.org/datasets/catalog/imdb_reviews

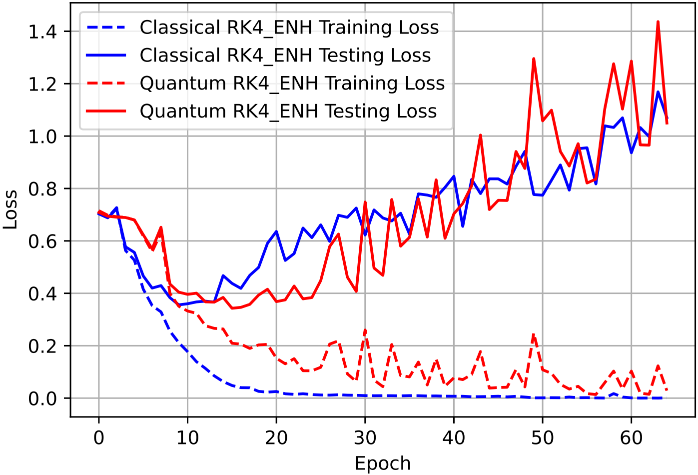

Figure: Cross-entropy loss evolution during learning

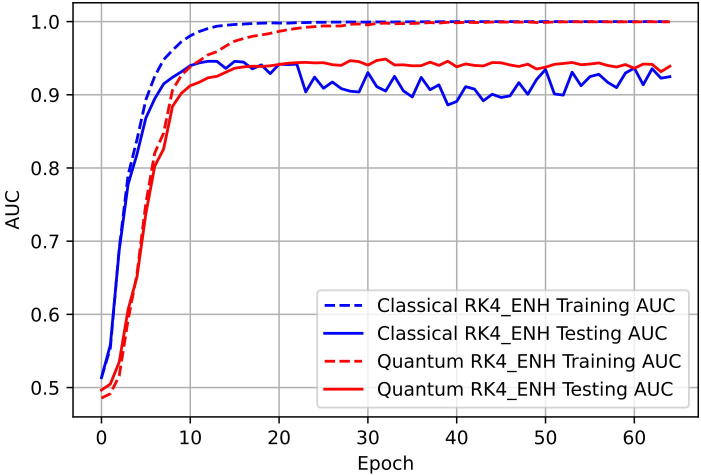

Figure: AUC score evolution during learning

Configurations:

Configurations:

- **Max sequence length**: 512
- **Classical / Quantum Hidden Size**: 12 / 6
- **Classical / Quantum ODE-Transformer Blocks**: 1 / 1
- **Classical / Quantum Attention Heads**: 6 / 2
- **Classical / Quantum Hidden MLP Size**: 6 / 4

Configurations:

- **Max sequence length**: 512
- **Classical / Quantum Hidden Size**: 12 / 6
- **Classical / Quantum ODE-Transformer Blocks**: 1 / 1
- **Classical / Quantum Attention Heads**: 6 / 2
- **Classical / Quantum Hidden MLP Size**: 6 / 4

ODE	Train Time (s)	Accuracy	F1 Score	Best AUC Epoch	# Parameters	# Qubits
RK4_ENH	**3328.61**	**85%**	**85%**	**13**	499316	-
QRK4_ENH	9033.13	**85%**	**85%**	33	**243896**	141

Table: IMDb metrics for the classical and quantum configurations

Synthetic Data Generation

Synthetic Data Generation

Metric	QRKT-GAN	TransGAN
Inception Score (IS)	74.89	52.31
Fréchet Inception Distance (FID)	66.78	46.97

Table: Performance Metrics for TransGAN and QRKT-GAN on CIFAR-10

Synthetic Data Generation

Metric	QRKT-GAN	TransGAN
Inception Score (IS)	74.89	52.31
Fréchet Inception Distance (FID)	66.78	46.97

Table: Performance Metrics for TransGAN and QRKT-GAN on CIFAR-10

Figure: Generated Images using QRKT-GAN

Technologies used for QRKT-GAN:

Technologies used for QRKT-GAN:

- JAX [17] and Flax [18]
- Tensorflow Quantum [19]
- Qiskit [20]
- Pytorch [21]
- Tensor Circuit [22]

Conclusion

Keywords:

Conclusion

Keywords:

- Deep Learning
- Transformers
- GANs
- Runge-Kutta
- Quantum
- Optimization

Conclusion

Keywords:

- Deep Learning
- Transformers
- GANs
- Runge-Kutta
- Quantum
- Optimization

$$|\text{Thank you!}\rangle$$

References

[1] Yun He, Ziwei Zhu, Yin Zhang, Qin Chen, and James Caverlee.
Infusing disease knowledge into bert for health question answering, medical inference and disease name recognition.
arXiv preprint arXiv:2010.03746, 2020.

[2] Petro Liashchynskyi and Pavlo Liashchynskyi.
Grid search, random search, genetic algorithm: a big comparison for nas.
arXiv preprint arXiv:1912.06059, 2019.

References (2)

[3] M. Hammad Hassan.
Random search.
https://medium.com/@hammad.ai/
tuning-model-hyperparameters-with-random-search-f4c1cc
2023.

[4] Jose M Alvarez and Mathieu Salzmann.
Learning the number of neurons in deep networks, 2018.

[5] Long Wen, Liang Gao, Xinyu Li, and Bing Zeng.
Convolutional neural network with automatic learning rate scheduler for fault classification.
IEEE Transactions on Instrumentation and Measurement, 70:1–12, 2021.

References (3)

[6] Alireza Azadbakht, Saeed Reza Kheradpisheh, Ismail Khalfaoui-Hassani, and Timothée Masquelier.
Drastically reducing the number of trainable parameters in deep cnns by inter-layer kernel-sharing.
arXiv preprint arXiv:2210.14151, 2022.

[7] Juyong Kim, Yookoon Park, Gunhee Kim, and Sung Ju Hwang.
SplitNet: Learning to semantically split deep networks for parameter reduction and model parallelization.
In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*,

References (4)

volume 70 of *Proceedings of Machine Learning Research*, pages 1866–1874. PMLR, 06–11 Aug 2017.

[8] Basheer Qolomany, Majdi Maabreh, Ala Al-Fuqaha, Ajay Gupta, and Driss Benhaddou.
Parameters optimization of deep learning models using particle swarm optimization.
In *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 1285–1290, 2017.

References (5)

[9] BRYON MOYER.
Generative adversarial network (gan).
https://semiengineering.com/knowledge_centers/artificial-intelligence/neural-networks/generative-adversarial-network-gan/, 2021.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin.
Attention is all you need.
Advances in neural information processing systems, 30, 2017.

References (6)

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby.
An image is worth 16x16 words: Transformers for image recognition at scale.
arXiv preprint arXiv:2010.11929, 2020.

[12] Yaofeng Desmond Zhong, Tongtao Zhang, Amit Chakraborty, and Biswadip Dey.
A neural ode interpretation of transformer layers.
arXiv preprint arXiv:2212.06011, 2022.

References (7)

[13] Yiping Lu, Zhuohan Li, Di He, Zhiqing Sun, Bin Dong, Tao Qin, Liwei Wang, and Tie-Yan Liu.
Understanding and improving transformer from a multi-particle dynamic system point of view.
arXiv preprint arXiv:1906.02762, 2019.

[14] Subhabrata Dutta, Tanya Gautam, Soumen Chakrabarti, and Tanmoy Chakraborty.
Redesigning the transformer architecture with insights from multi-particle dynamical systems.
Advances in Neural Information Processing Systems, 34:5531–5544, 2021.

References (8)

[15] Marçal Comajoan Cara, Gopal Ramesh Dahale, Zhongtian Dong, Roy T. Forestano, Sergei Gleyzer, Daniel Justice, Kyoungchul Kong, Tom Magorsch, Konstantin T. Matchev, Katia Matcheva, and Eyup B. Unlu.
Quantum vision transformers for quark-gluon classification.
Axioms, 13(5):323, May 2024.

[16] Yifan Jiang, Shiyu Chang, and Zhangyang Wang.
Transgan: Two pure transformers can make one strong gan, and that can scale up.
Advances in Neural Information Processing Systems, 34:14745–14758, 2021.

References (9)

[17] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, et al.
Jax: composable transformations of python$+$ numpy programs.
2018.

[18] Jonathan Heek, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas Steiner, and Marc van Zee.
Flax: A neural network library and ecosystem for jax.
Version 0.3, 3:14–26, 2020.

References (10)

[19] Michael Broughton, Guillaume Verdon, Trevor McCourt, Antonio J Martinez, Jae Hyeon Yoo, Sergei V Isakov, Philip Massey, Ramin Halavati, Murphy Yuezhen Niu, Alexander Zlokapa, et al.
Tensorflow quantum: A software framework for quantum machine learning.
arXiv preprint arXiv:2003.02989, 2020.

[20] Andrew Cross.
The ibm q experience and qiskit open-source quantum computing software.
In *APS March meeting abstracts*, volume 2018, pages L58–003, 2018.

References (11)

[21] Sagar Imambi, Kolla Bhanu Prakash, and GR Kanagachidambaresan.
Pytorch.
Programming with TensorFlow: Solution for Edge Computing Applications, pages 87–104, 2021.

[22] Shi-Xin Zhang, Jonathan Allcock, Zhou-Quan Wan, Shuo Liu, Jiace Sun, Hao Yu, Xing-Han Yang, Jiezhong Qiu, Zhaofeng Ye, Yu-Qin Chen, Chee-Kong Lee, Yi-Cong Zheng, Shao-Kai Jian, Hong Yao, Chang-Yu Hsieh, and Shengyu Zhang.
Tensorcircuit: a quantum software framework for the nisq era.
Quantum, 7:912, February 2023.