

# YouroQNet

Quantum Text Classification with Context Memory

Team: QwQ

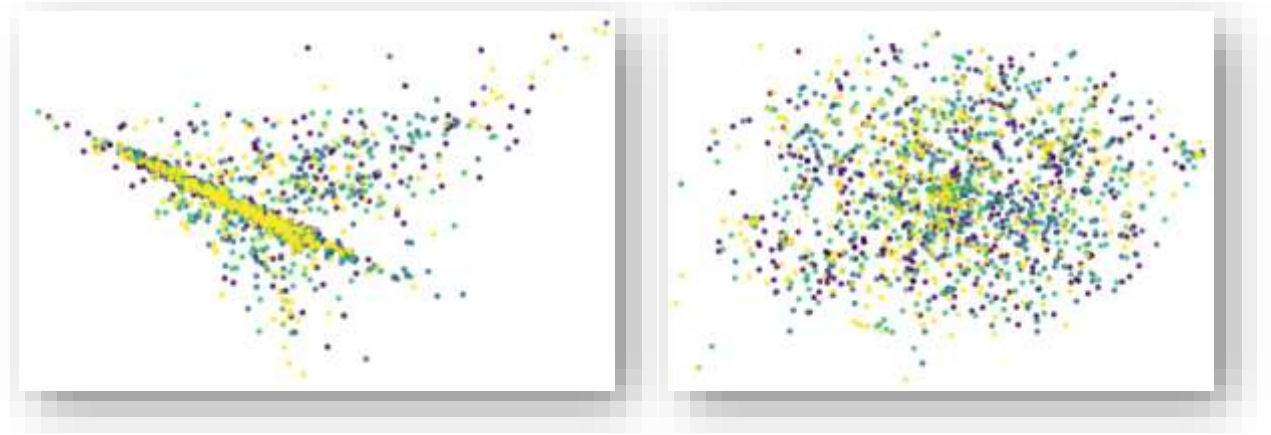
Reporter: Armit

# Text Classification in a Quantum Manner

- what the problem is
  - NLP sentiment comprehension
  - the dataset just holy shit
- how current methods work
  - TextCNN, TextRNN, BERT
  - QNLP, QSANN
- nevertheless our contributions
  - heuristic k-gram tokenizer
  - YouroQNet
  - computational analysis

```
label,text
0,好戏开锣了
0,可是她其实爱着他。
0,总之就是很像我觉得有胡子更像知道宋山木为啥一直留胡子吗？因为没胡子的宋山木确实很像强奸犯。（被揭
0,豪犀利啊~ 孩子他妈。
0,好心大了，所有的大事都小了。【禅语悟道】心小了，所有的小事就大了；心大了，所有的大事都小了。大事
0,“你说哪个？强哥结婚了，你没机会了哇！强哥你结婚了啊~我老婆是狮子座 狮子：不顾形象的家长处女：变
0,我焦急地凝望南方已五千年！夜色有些宁静也有些冷清，寒风吹过我的神智突然好清醒，此时真希望这只是一
0,300+600+1200+是什么时候开始流行的哇塞~ 北京的书啥时候来啊~ 2010~ 2011年秋冬潮流专刊今日上市了
0,一起度过我们在一起的第一个生日吧！亲口对他说一句生日快乐。温州昆明巡演订票 搞橙子方法请见另，个人
0,看球了看球了。看这种比赛就是纯欣赏了，反正晋级了多半也要被魔兽给兽兽掉。
0,“等我找到男朋友，我第一件事就是抽他两嘴巴。”众惊，忙问缘由。“我得问问，这些年你tmd 躲哪去了！”
0,与元稿子了，我终于可以把老师从黑名单里放出来了，万豪科学总是当月给稿费！万豪科学是本儿拜谱杂志！
```

ambiguous example, wrong label and informal pragmatics



PCA & TSNE over word-level TF-IDF

这/是/一/个/例/子	$\log p = -50.4$
这是/一个/例子	$\log p = -46.4$
这是/一个/例/子	$\log p = -39.0$

这/是/一/个/例/子	logp = -50.4
这是/一个/例子	logp = -46.4
这是/一个/例子	logp = -39.0

• • • • •

Diagram illustrating a word embedding matrix structure:

- A large rectangle represents the embedding space.
- The top-left corner is labeled "PAD" and "我是一个字典..." (I am a dictionary...), indicating a padding token.
- The bottom-right area is labeled "D=16" (embedding dimension) and "K=3000+" (vocabulary size).
- Inside this area, the text  $\theta_j$  and "....." are shown, representing the embedding vector for a specific word.

Diagram illustrating a word embedding matrix structure:

- A large rectangle represents the embedding space.
- The top-left corner is labeled "PAD" and "我是一个字典..." (I am a dictionary...).
- The bottom-right area is labeled "D=16" and "K=3000+".
- Inside the bottom-right area, the text  $\theta_j$  and "....." are shown.

(split or pad to model length)

(split or pad to model length)

这 / 是 / 一个  
是 / 一个 / 例子

(binary cross entropy)

(binary cross entropy)

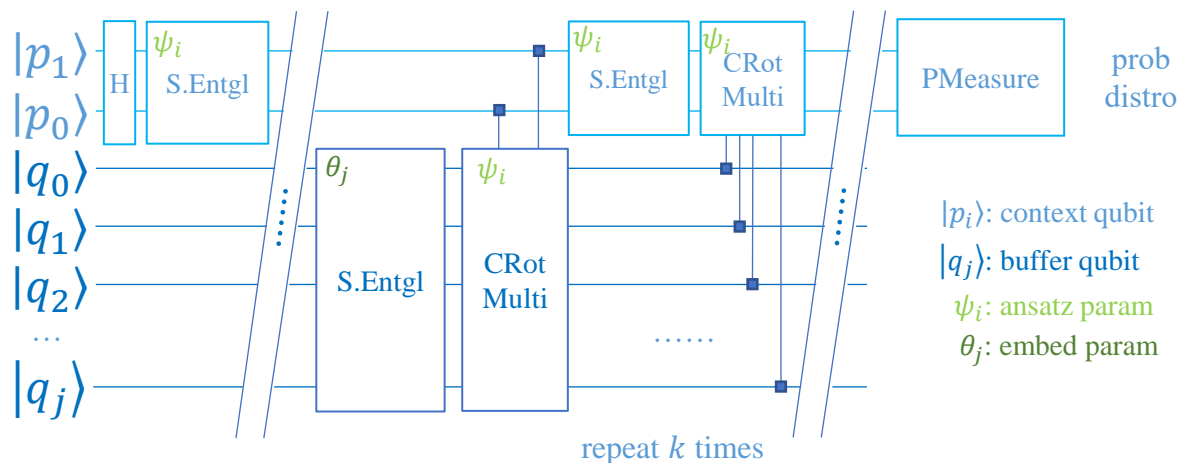
$$= y \log(x) + (1 - y) \log(1 - x)$$

 $x \in [0,1], y \in \{0,1\}$ 

The diagram illustrates the process of converting a sentence into a numerical representation for a neural network. It starts with the sentence "这是一个例子" (This is an example). This sentence is then "tokenize" (tokenized) into the sequence "这 / 是 / 一 / 个 / 例 / 子". These tokens are then "embed" (embedded) into a vector space, represented as  $[L=4, D]$ . A green starburst labeled "learnable" points to this embedding step. This vector is then "align" (aligned) with another vector  $[mL=3, D]$ . The two vectors are then combined using an "ansatz" (ansatz) operation, resulting in a new vector  $[nq^2=16]$ . This vector is then "project" (projected) into a final vector space  $[NC=4]$ . Finally, a "loss" (loss) is calculated between the final vector and a target vector  $[0, 1, 0, 0]$ , which is labeled as a "(onehot label)".

(YouroQNet)

```
n_qubit = model_length + n_class
```



Read the circuit:

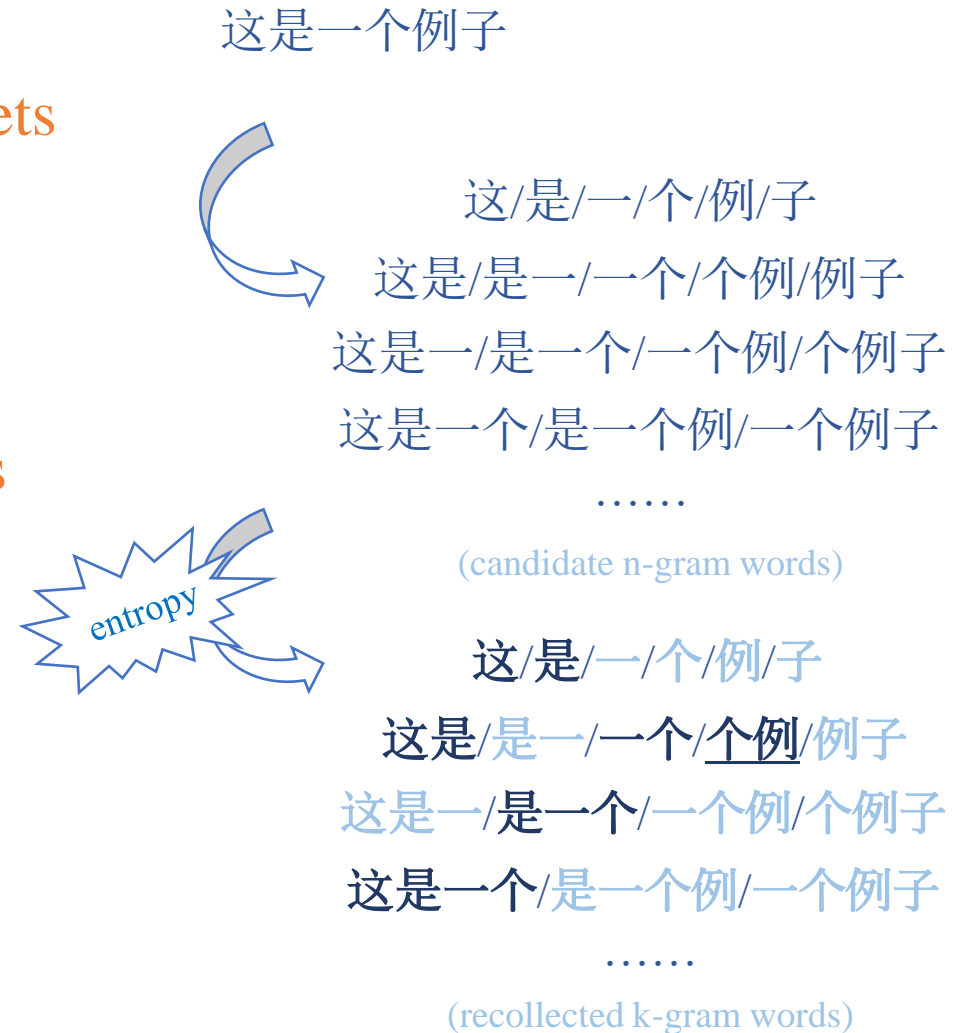
- $|p_i\rangle$  is current context,  $|q_j\rangle$  is buffer for incoming data
- param  $\psi_i$  is for ansatz,  $\theta_j$  is from word embed
- initialize context with H gate
- bias context via a SEC (Strong Entangle Circuit)
- repeat k times for sequence comprehension
  - load data via a SEC
  - write to context via a CMC (CRot Multi Circuit)
  - digest context via a SEC
  - read from context via a CMC
- prob measure on context qubits

# Content Table

- Entropy-based Heuristic k-gram Tokenizer
- YouroQNet for Quantum Sequence Classification
- Computational Analysis over Simple QCircuits

# Entropy-based Heuristic k-gram Tokenizer

- why tokenizing
  - not necessary, even not work on small datasets
  - reduce input length for limited qubits
- from fixed n-gram to adaptive k-gram
  - build several n-gram vocabs from the corpus
    - word presence probability
  - tokenize corpus by longest match
    - bidirectional beam search
  - recollect all tokenized words as new vocab
    - can be iterative if needed



# Tokenizer: concrete example

- processing pipeline
  - text normalization
  - make char and 2/3/4/5-gram vocab
    - filter Chinese words for n-gram
    - truncate size by min\_freq=3
    - calculate word presence probability
  - tokenize the corpus with merged vocabs
    - build trie tree for longest match
    - bi-directional beam search with n\_beam=3
    - gather all tokenized tokens
    - truncate size by min\_freq=5
  - maximize  $p(W) = \prod_i p(w_i)$ 
    - $p(\text{这})p(\text{是})p(\text{一})p(\text{个例})p(\text{子})$
    - $p(\text{这是})p(\text{一个})p(\text{例})p(\text{子})$
    - ...

```
s = R_DEADCHAR.sub('', s)
s = R_CJK_PERIOD.sub('.', s)
s = R_CJK_PUASE.sub(',', s)
s = wchar_to_char(s)
s = s.lower()
s = R_WHITESPACE.sub(' ', s)
s = R_NUMBER.sub('0', s)
s = try_concat(s)
s = fold_triple(s)
```

text norm

n-gram	n_vocab	n_vocab trunc.
char	3345	-
2-gram	34363	4438
3-gram	48720	1264
4-gram	45478	238
5-gram	38812	44
k-gram	4934	2713
k-gram+	6376	2899

n/k-gram vocabs size

1 , 3822	一个 201	一个人 58	心不开心 32	开心不开心 32
2 . 2895	我们 176	的时候 55	开心不开 32	不开心不开 31
3 的 2717	什么 158	不知道 39	不开心不 31	心不开心不 30
4 我 1587	不是 125	不开心 37	分享图片 17	黑板上画了 5
5 ! 1566	我的 121	为什么 33	我不知道 10	在黑板上画 5
6 了 1437	就是 119	心不开 32	内牛满面 10	不知道什么 5
7 是 1290	这个 118	开心不 32	什么时候 10	知道什么叫 4
8 不 1212	自己 115	自己的 31	生日快乐 9	爱情就像便 4
9 一 1136	哈哈 105	这样的 27	黑板上画 7	有机会获得 4
10 0 983	今天 103	是一个 24	让我们再 7	情就像便便 4
11 人 789	的人 102	是不是 23	新浪微博 7	发光的友情 4
12 ~ 783	大家 100	分享图 22	不是因为 7	这样的男人 3
13 你 735	真的 96	演唱会 21	这就是我 6	这就是传说 3
14 有 730	回复 91	一定要 21	转发微博 6	还在黑板上 3
15 这 681	没有 89	真的很 19	知道什么 6	观世音菩萨 3

n-gram vocabs

# Tokenizer: interactive demo

```
C:\Windows\System32\cmd.exe - conda activate q - python.exe vis_tokenizer.py
-> py vis_tokenizer.py
input a sentence: 这是一个例子
[-39.038] 这是一个例子
[-43.846] 这是一个例子
[-50.386] 这是一个例子
input a sentence: 这是另一个例子
[-52.893] 这是另一个例子
[-53.912] 这是另一个例子
[-59.433] 这是另一个例子
input a sentence: 好多例子啊
[-39.447] 好多例子啊
[-44.754] 好多例子啊
[-56.207] 好多例子啊
input a sentence: 南京市长江大桥
[-49.046] 南京市长江大桥
[-58.014] 南京市长江大桥
input a sentence: 微博新闻有什么好看的
[-42.166] 微博新闻有什么好看的
[-46.414] 微博新闻有什么好看的
input a sentence: 半自动分词器
[-46.861] 半自动分词器
[-56.745] 半自动分词器
input a sentence: 如果有更多的数据就好了
[-54.971] 如果有更多的数据就好了
[-60.052] 如果有更多的数据就好了
[-67.515] 如果有更多的数据就好了
input a sentence: _
```

```
C:\Windows\System32\cmd.exe - conda activate q - python.exe vis_tokenizer.py
-> py vis_tokenizer.py
input a sentence: 如果有标点符号，会怎么样呢？
[-81.634] 如果有标点符号，会怎么样呢？
[-81.651] 如果有标点符号，会怎么样呢？
[-87.659] 如果有标点符号，会怎么样呢？
input a sentence: 甚至：有一些神秘空格》》和？符号
[-98.282] 甚至：有一些神秘空格》》和？符号
[-103.860] 甚至：有一些神秘空格》》和？符号
[-113.051] 甚至：有一些神秘空格》》和？符号
input a sentence: 全角符号。，会变成ASCII码的版本
[-121.670] 全角符号。，会变成ASCII码的版本
[-128.043] 全角符号。，会变成ASCII码的版本
input a sentence: 当然，英文iphone会被拆得很奇怪
[-105.886] 当然，英文iphone会被拆得很奇怪
[-106.820] 当然，英文iphone会被拆得很奇怪
[-107.658] 当然，英文iphone会被拆得很奇怪
[-113.868] 当然，英文iphone会被拆得很奇怪
input a sentence: 不不不不要停下来啊！！！！！！单字重复会被缩略
[-118.065] 不不要停下来啊！！单字重复会被缩略
[-123.814] 不不要停下来啊！！单字重复会被缩略
[-123.962] 不不要停下来啊！！单字重复会被缩略
input a sentence: 这就是关于我的一切 :)
[-59.600] 这就是关于我的一切 :)
[-65.409] 这就是关于我的一切 :)
[-65.785] 这就是关于我的一切 :)
input a sentence: _
```

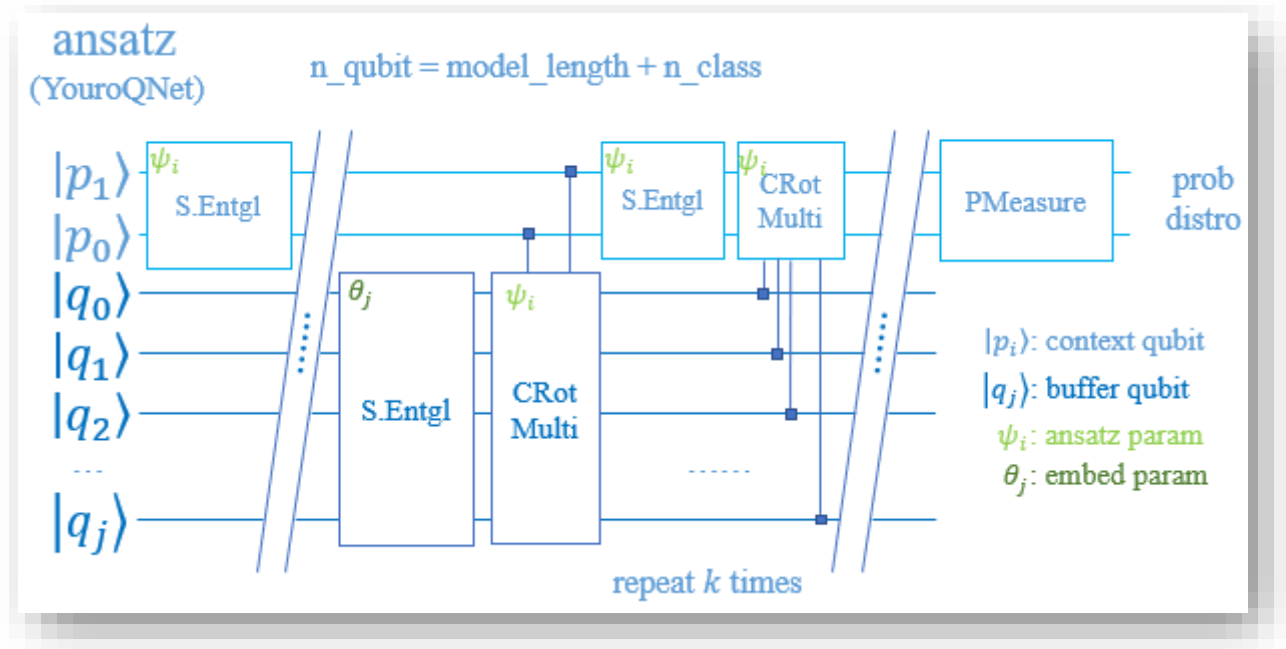
# Content Table

- Entropy-based Heuristic k-gram Tokenizer
- **YouroQNet for Quantum Sequence Classification**
- Computational Analysis over Simple QCircuits



# YouroQNet for Quantum Sequence Classification

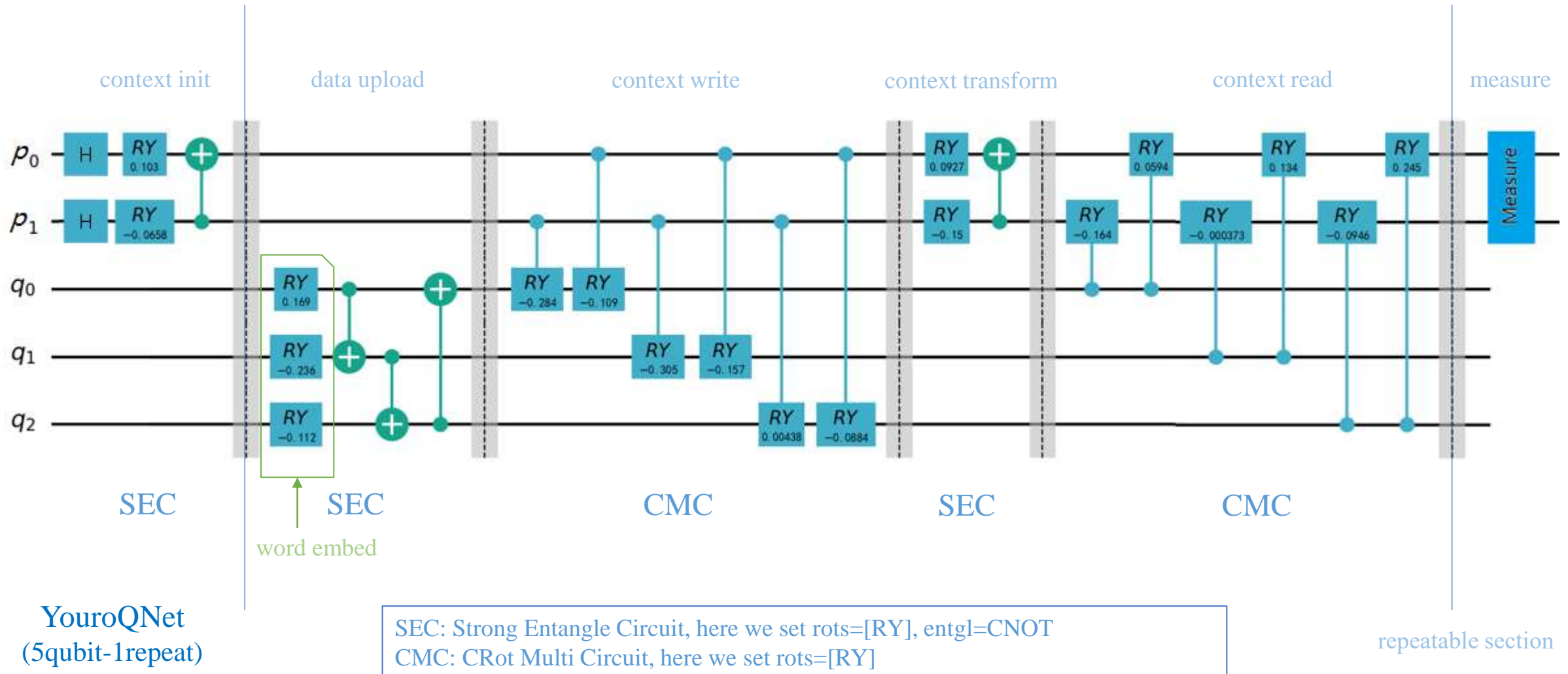
- YouroQNet :=
  - pure variational quantum circuit
  - representation learning
  - style-transfer scheme
- what is for
  - sequence classification
  - feature abstraction



YouroQNet general architecture

# YouroQNet: minimal concrete example

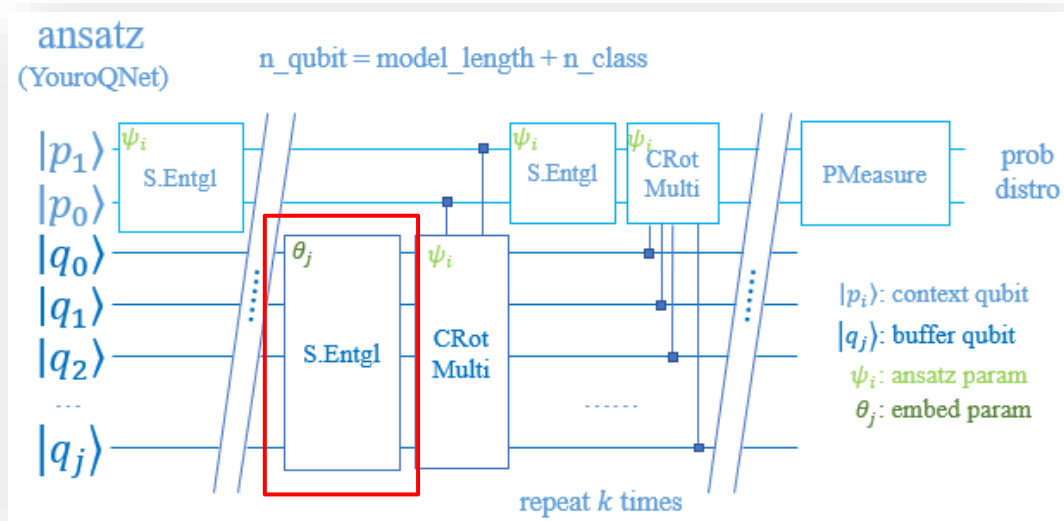
- pure variational quantum circuit



# YouroQNet learns semantical word representation

- YouroQNet :=
  - pure variational quantum circuit
  - **representation learning**
  - implicit convolutional
  - implicit recurrent
  - style-transfer scheme

Train the word embedding  $\theta_j$  whose channels are partially temporally dependent.

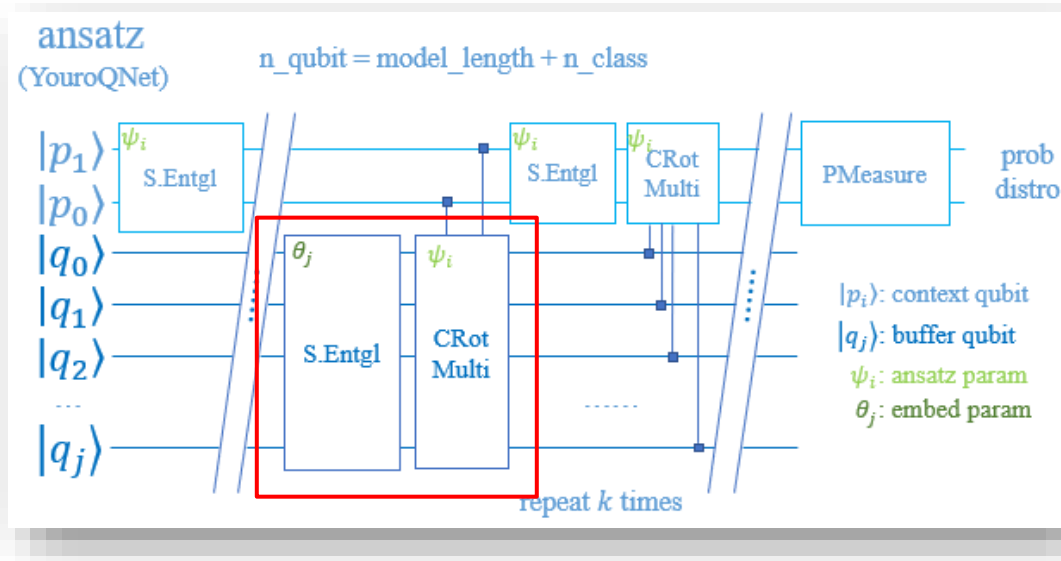


# YouroQNet differs contextual invariants from variants

- YouroQNet :=
  - pure variational quantum circuit
  - **representation learning**
  - implicit convolutional
  - implicit recurrent
  - style-transfer scheme

Train  $\psi_i$  and  $\theta_j$  alternatively  
alike the EM algorithm.

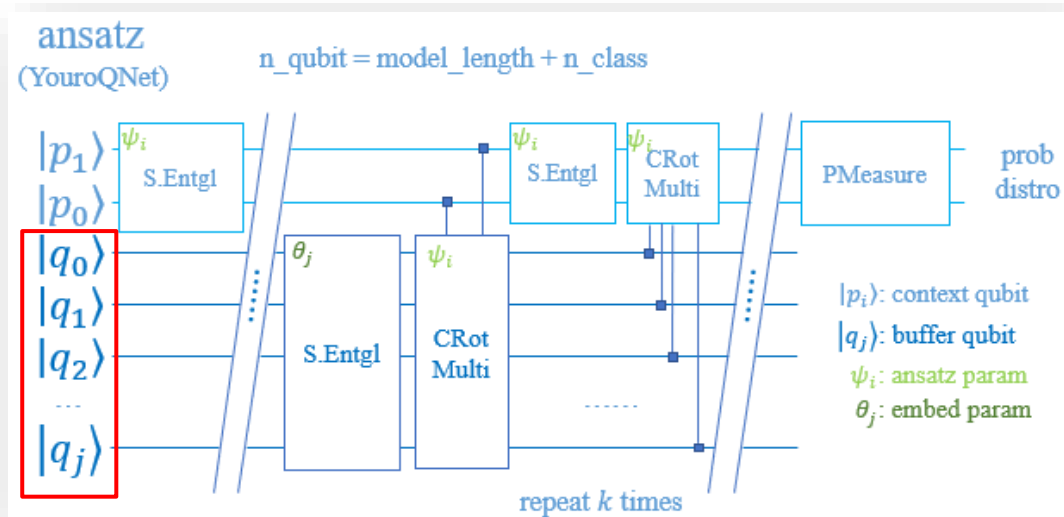
Note that  $\psi_i$  is constant to all  
inputs, corresponding to  
contextual invariant (syntactical)  
transforms, while  $\theta_j$  builds up a  
(semantical) variant context.



# YouroQNet is implicitly convolutional

- YouroQNet :=
  - pure variational quantum circuit
  - representation learning
  - **implicit convolutional**
  - implicit recurrent
  - style-transfer scheme

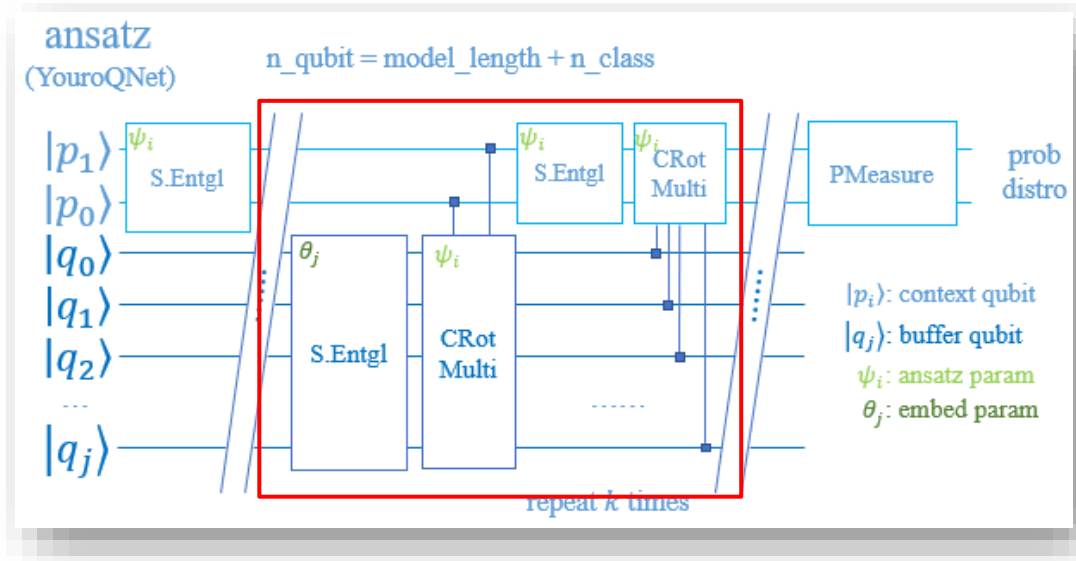
Breaking down long sentences with the **aligner**, a YouroQNet is like a single Conv1d filter that slides along the sentence then applies an AvgPooling.



# YouroQNet could be recurrent if needed

- YouroQNet :=
  - pure variational quantum circuit
  - representation learning
  - implicit convolutional
  - **implicit recurrent**
  - style-transfer scheme

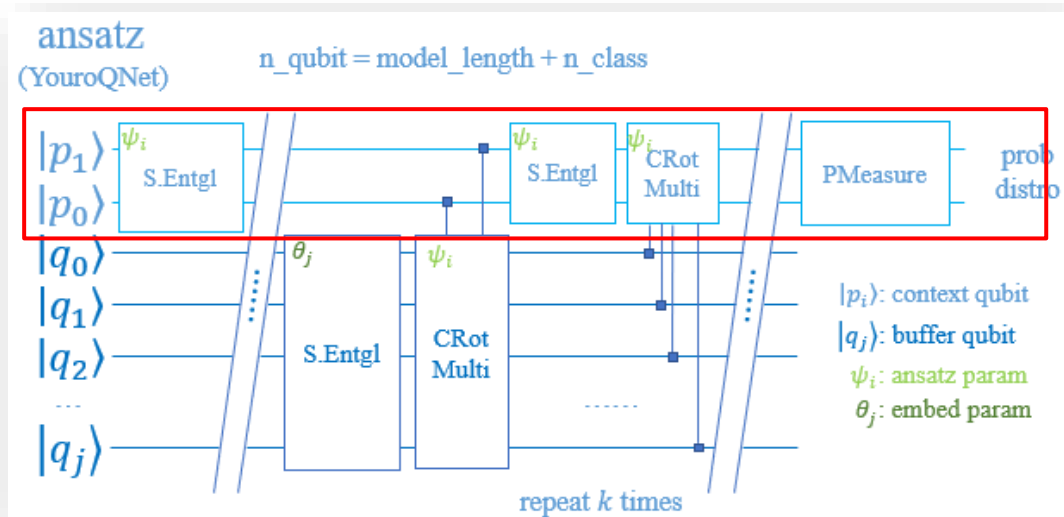
The ancilla  $|p_i\rangle$  is like a state memory, when  $\psi_i$  is shared across all repeatable parts of the circuit, it simulates an RNN.



# YouroQNet follows a style-transfer scheme

- YouroQNet :=
  - pure variational quantum circuit
  - representation learning
  - implicit convolutional
  - implicit recurrent
  - **style-transfer scheme**

The ancilla  $|p_i\rangle$  is the blank canvas, we gradually extract useful info from  $|q_j\rangle$  and transfer onto it, through steps.



# YouroQNet: metric evaluation & comparison

F1 score	YouroQNet	Text DNN	Text CNN	Text RNN	TF-IDF					FastText word2vec				
					kNN	GBDT	Bayes	SVM	MLP	kNN	GBDT	Bayes	SVM	MLP
Joy	<u>0.018</u>	0.349	0.321	0.287	0.338	0.214	0.357	0.374	0.41	0.361	0.413	0.430	0.458	0.455
Angry	<u>0.342</u>	0.294	0.175	0.32	0.327	0.356	0.300	0.336	0.291	0.347	0.333	0.375	0.393	0.370
Sad	<u>0.22</u>	0.306	0.321	0.283	0.175	0.169	0.274	0.349	0.306	0.310	0.24	0.171	0.328	0.225
Hate	<u>0.273</u>	0.187	0.275	0.318	0.271	0.259	0.308	0.24	0.313	0.179	0.289	0.306	0.314	0.290
Avg.	<u>0.213</u>	0.284	0.273	0.302	0.278	0.25	0.31	0.325	0.33	0.3	0.319	0.321	0.373	0.335



# YouroQNet - mini: toy verification in details

- toy model config
  - 3+1 qubits, 1 repeat, [RY]-CNOT-[RY]
- embedding analysis
  - manually bias the dataset
  - embed highlights the minority
  - 👉 it learns TF-IDF successfully !!

```
words = {  
    # positive (leading to class 0)  
    '爱', '喜欢',  
    # negative (leading to class 1)  
    '恨', '讨厌',  
    # neutral  
    '啊', '我', '你', '苹果', '西瓜',  
}
```

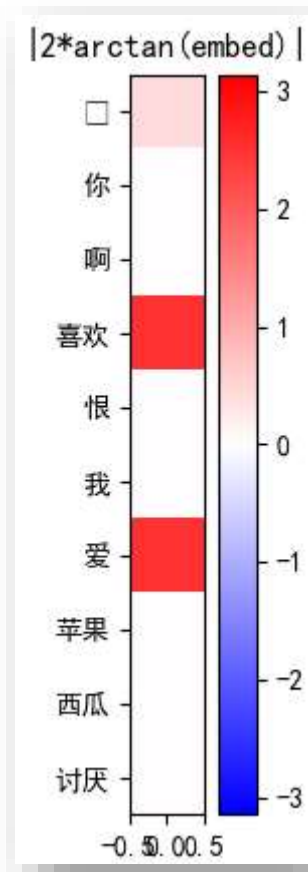
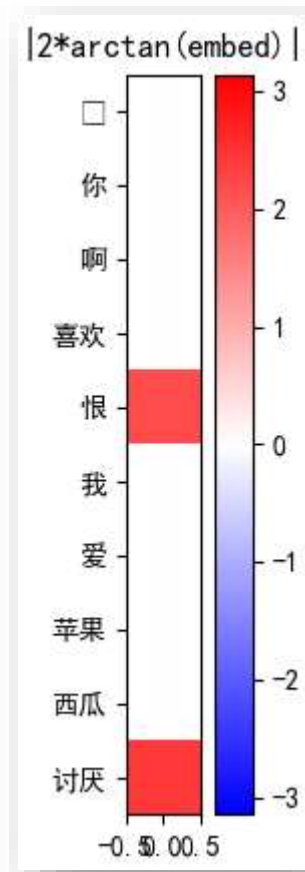
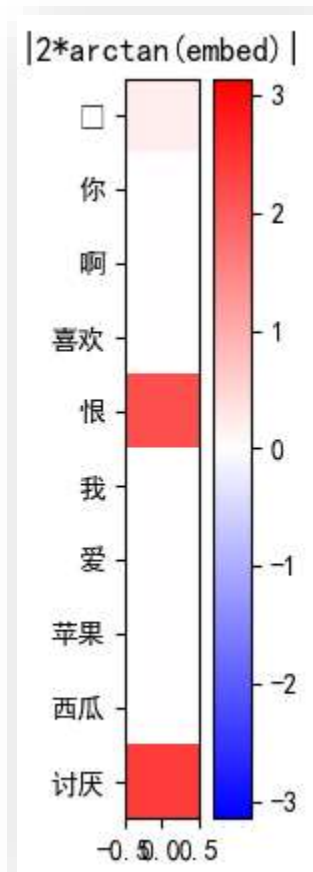
vocab

```
train_data = [  
    (0, '我爱你'),  
    (0, '我喜欢苹果'),  
    (0, '苹果啊喜欢'),  
    (0, '你爱西瓜'),  
    (1, '你讨厌我'),  
    (1, '讨厌西瓜苹果'),  
    (1, '你讨厌苹果'),  
    (1, '我恨啊恨'),  
] + biased_data
```

train data

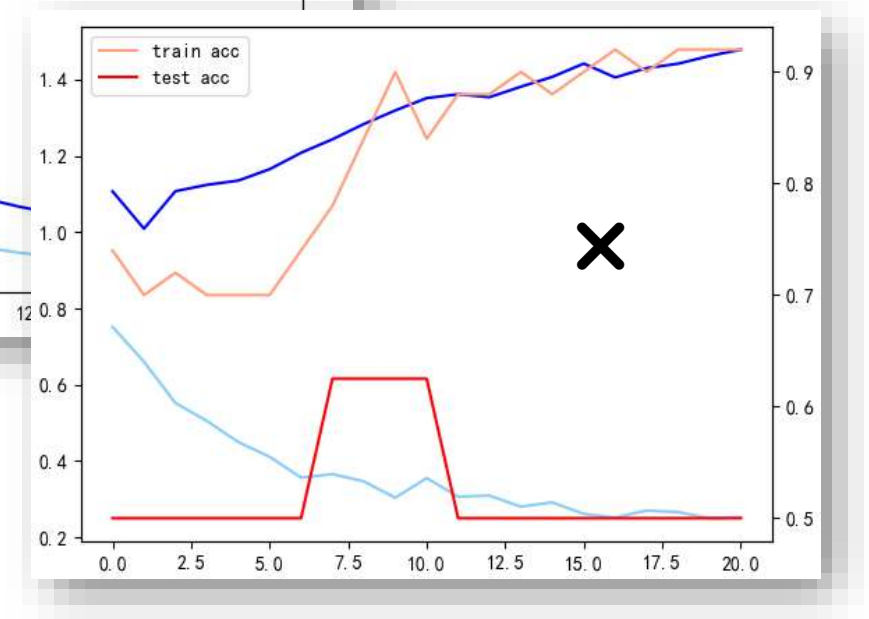
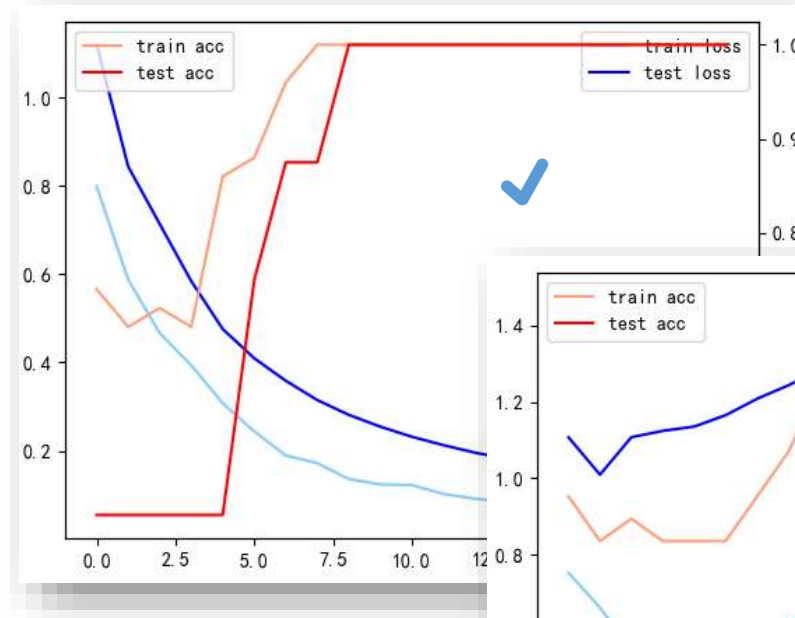
```
biased_data = [  
    (0, '我你'),  
    (0, '你啊'),  
    (0, '啊我啊啊'),  
    (0, '西瓜苹果'),  
    (0, '你西瓜'),  
    (0, '我苹果啊'),  
]
```

pos = 0  
neg = 1



# YouroQNet: QNN is fragile | symmetric, periodical, finite-valued

- parameter initialization
  - uniform ✗
  - normal ✓
- embedding normalization
  - $\pm \pi/2$  ✗
  - $\pm \pi$  ✓
- gradient method
  - param\_shift ✗
  - finite\_diff ✓
- loss not decay or quickly overfit
  - tune rand seed
  - kill & retry 🤖



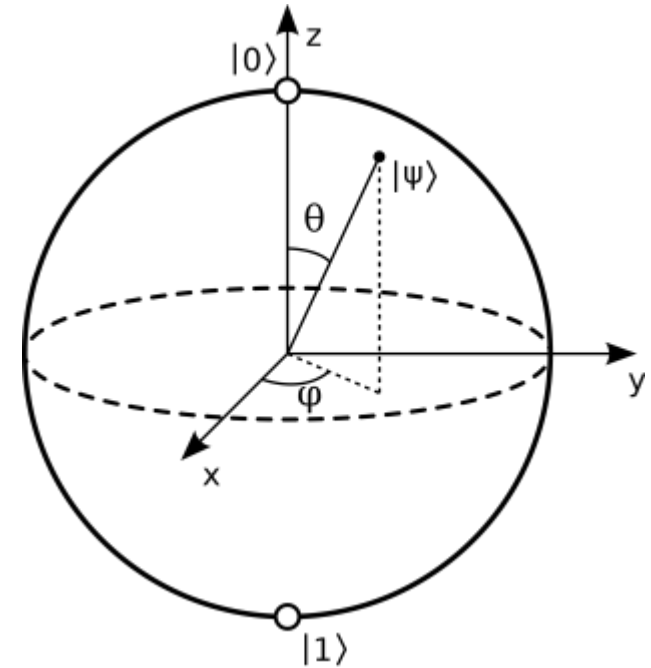
good & bad loss curves

# Content Table

- Entropy-based Heuristic k-gram Tokenizer
- YouroQNet for Quantum Sequence Classification
- Computational Analysis over Common QCircuits

# Rethink on QCircuit

- Conceptual model in meme
  - Balancer / 3D Clock
  - Map-Reduce
- But what's the mathematical model?
  - Single qubit rotation (superposition)
  - Multi-qubit entanglement

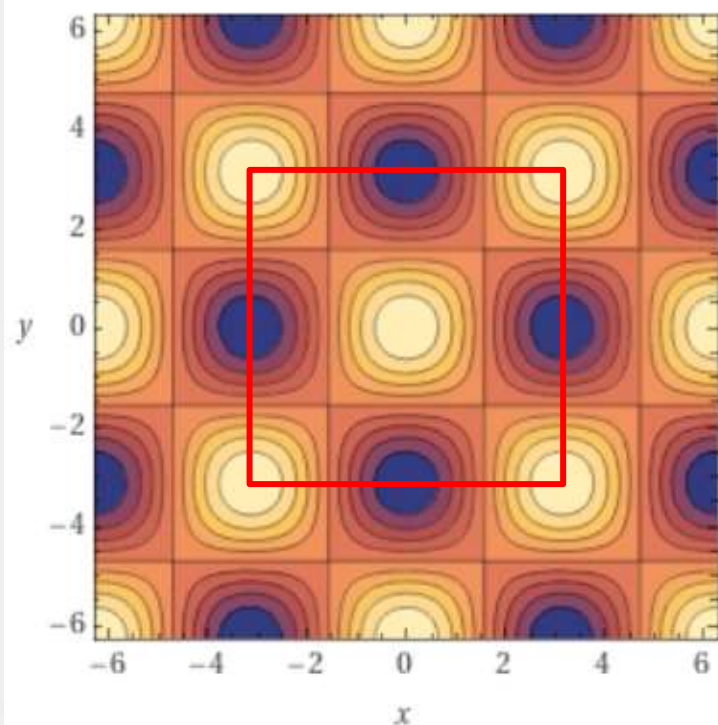


<https://leimao.github.io/blog/Qubit-Bloch-Sphere/>

# XYZ - RotCircuit

- numerical XOR!!

等高線プロット



```
# the single-qubit data encoder:
```

```
# - 1 qubit
```

```
# - 1~3 param
```

```
# - rotate with RX/RY/RZ
```

```
# - no entangle
```

```
# circuit:      c0      c1      c2
```

```
#  
# q_0: |0> → [RX(tht0)] → [RY(tht1)] → [RZ(tht2)] →  
#
```

```
# coeffs on:
```

```
# |0>: ( I*sin(θ_0/2)*sin(θ_1/2) + cos(θ_0/2)*cos(θ_1/2))*exp(-I*θ_2/2)
```

```
# |1>: (-I*sin(θ_0/2)*cos(θ_1/2) + sin(θ_1/2)*cos(θ_0/2))*exp( I*θ_2/2)
```

```
# probs on:
```

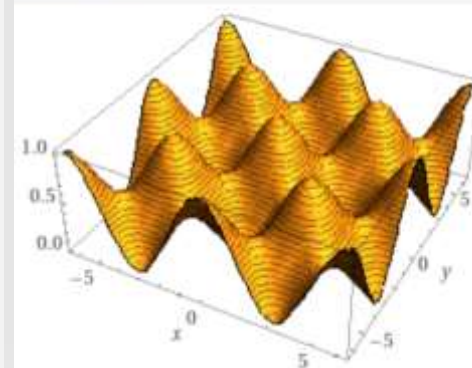
```
# |0>: 0.5*cos(θ_0)*cos(θ_1) + 0.5
```

```
# |1>: -0.5*cos(θ_0)*cos(θ_1) + 0.5
```

入力

$0.5 \cos(x) \cos(y) + 0.5$

3Dプロット



# YouroQNet - mini

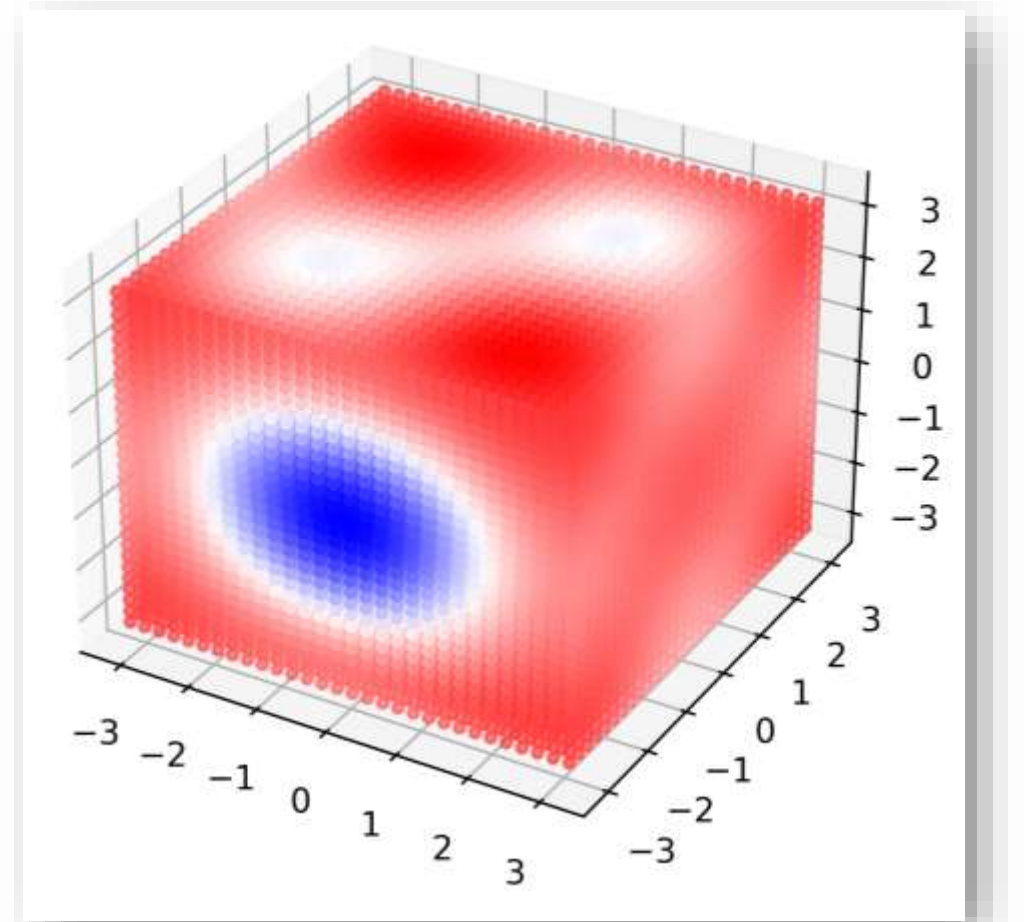
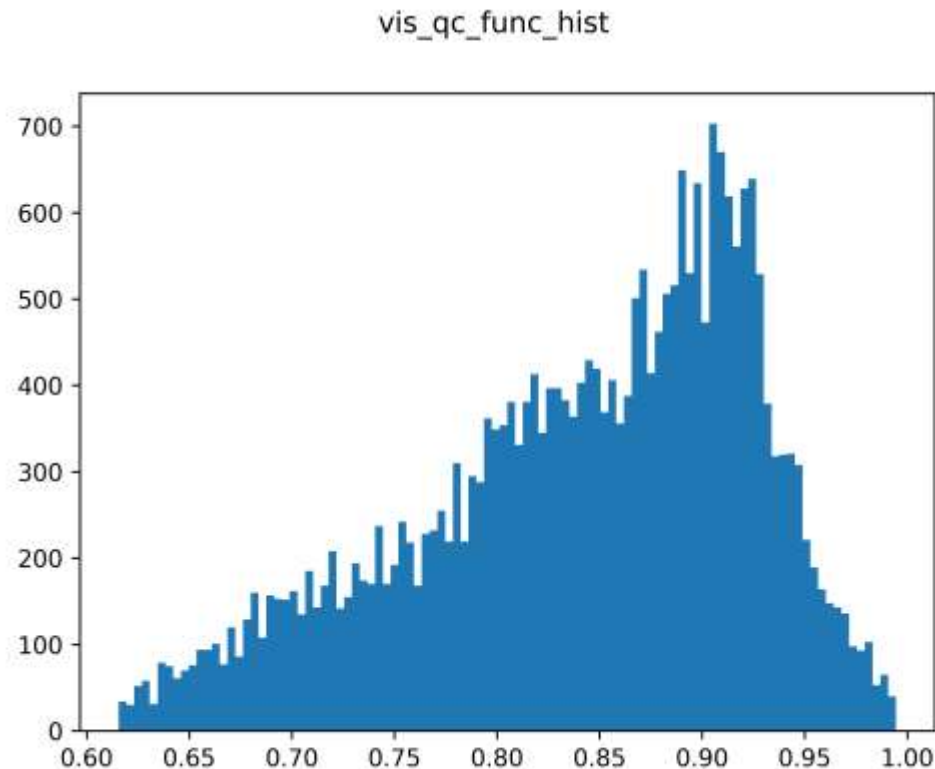
- combinational sum with trigonometric activation!!

```
# the YouroQNet toy (binary):
# - 4 qubits
# - 11 param (tht=3, psi=8)
# - rotate with RY
# - entangle with CNOT / CRY
# circuit:      c0      c1      c2      c3      c4      c5      c6      c7      c8      c9      c10
#
# q_0: |+> RY(psi0)
#
# q_1: |0> RY(tht0)
#
# q_2: |0> RY(tht1)
#
# q_3: |0> RY(tht2)
#
# coeff α on component α|0000> (eqv. matrix cell of qc[0, 0]):
# - sin(θ_0/2)*sin(θ_1/2)*sin(θ_2/2) * sin(ψ_0/2 + π/4) * cos(ψ_1/2 + ψ_2/2)*cos(ψ_3/2) * sin(ψ_4/2)
# + sin(θ_0/2)*sin(θ_1/2)*cos(θ_2/2) * sin(ψ_0/2 + π/4) * sin(ψ_1/2 + ψ_2/2)*sin(ψ_3/2) * sin(ψ_4/2)
# + cos(θ_0/2)*cos(θ_1/2)*sin(θ_2/2) * sin(ψ_0/2 + π/4) * sin(ψ_1/2 + ψ_2/2)*cos(ψ_3/2) * sin(ψ_4/2)
# - cos(θ_0/2)*cos(θ_1/2)*cos(θ_2/2) * sin(ψ_0/2 + π/4) * cos(ψ_1/2 + ψ_2/2)*sin(ψ_3/2) * sin(ψ_4/2)
# + sin(θ_0/2)*sin(θ_1/2)*sin(θ_2/2) * cos(ψ_0/2 + π/4) * cos(ψ_1/2 + ψ_2/2)*cos(ψ_3/2) * sin(ψ_4/2)
```

+ψ\_5/2 + ...

# YouroQNet - mini: learned function value space

- weird generalization...



Value space of a learned YouroQNet-mini  
(input is word embedding  $\theta$ )



# Content Table

- Entropy-based Heuristic k-gram Tokenizer
- YouroQNet for Quantum Sequence Classification
- Computational Analysis over Simple QCircuits

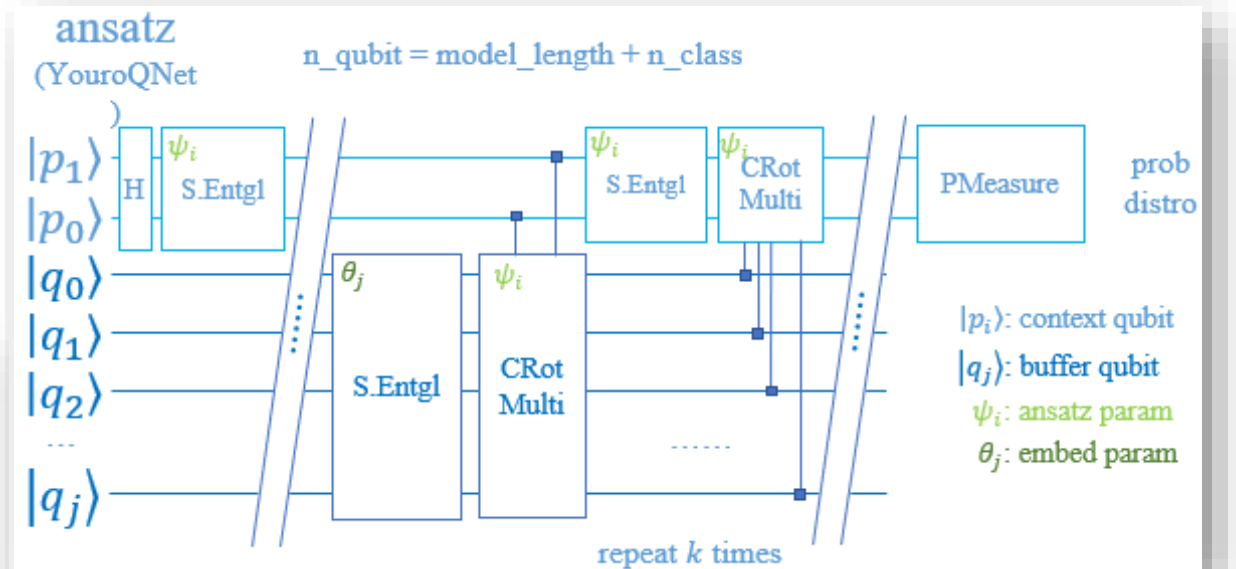
maximize  $p(W) = \prod_i p(w_i)$

- $p(\text{这})p(\text{是})p(\text{一})p(\text{个例})p(\text{子})$
- $p(\text{这是})p(\text{一个})p(\text{例})p(\text{子})$
- ...

atrix cell of qc[0, 0]):

```
sin(ψ_0/2 + pi/4) * cos(ψ_1/2 + ψ_2/2)*cos(ψ_3/2) * sin(ψ_4/2)
sin(ψ_0/2 + pi/4) * sin(ψ_1/2 + ψ_2/2)*sin(ψ_3/2) * sin(ψ_4/2)
sin(ψ_0/2 + pi/4) * sin(ψ_1/2 + ψ_2/2)*cos(ψ_3/2) * sin(ψ_4/2)
sin(ψ_0/2 + pi/4) * cos(ψ_1/2 + ψ_2/2)*sin(ψ_3/2) * sin(ψ_4/2)
cos(ψ_0/2 + pi/4) * cos(ψ_1/2 + ψ_2/2)*sin(ψ_3/2) * sin(ψ_4/2)
```

+ψ\_5/2 + ...





Thanks for your watching~

YouroQNet

熔炉ネットと言うのは、虚仮威し全て裏技を繋ぐもん

Team: QwQ

Reporter: Armit