

# Generative Adversarial Networks based Pseudo-Random Number Generator for Embedded Processors

<https://youtu.be/q5g6LER217M>

# Contents

**Generative Adversarial Network**

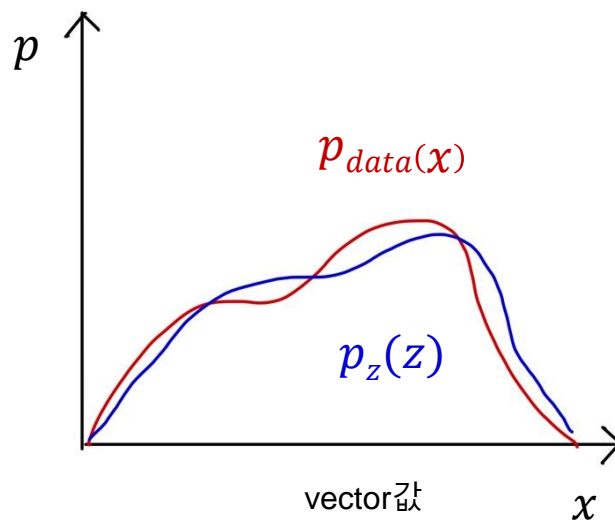
**Proposed Method**

**Evaluation**



CryptoCraft LAB

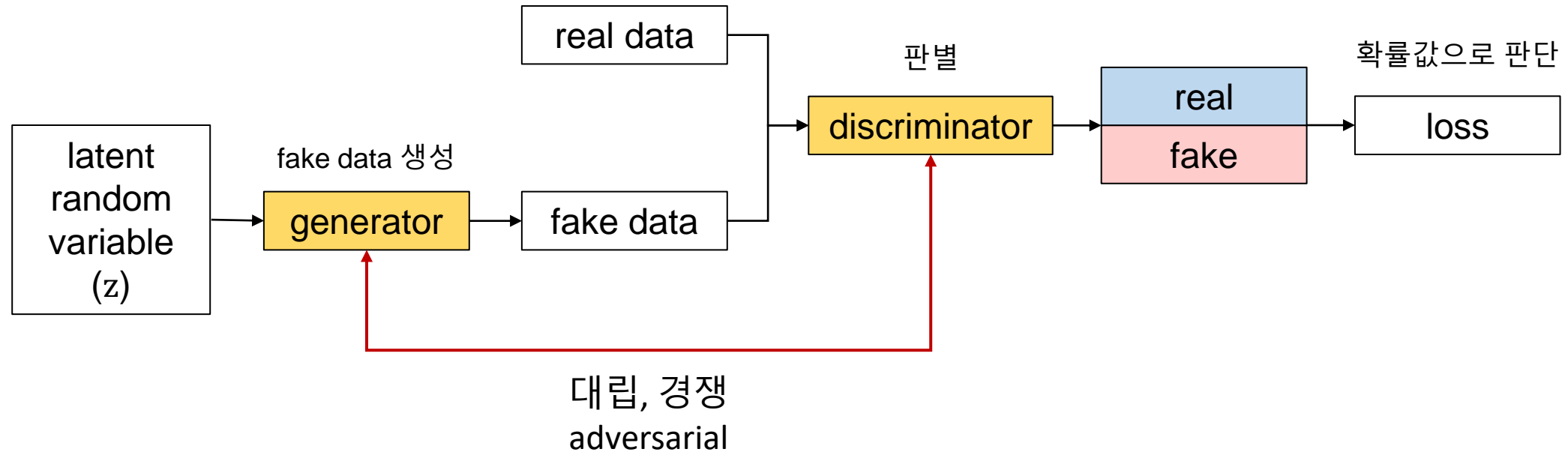
# Generative Adversarial Network



- data의 특징을 나타내는 vector값의 분포 ( $p$ )
- real data의 분포( $p_{data}(x)$ )와 fake data의 분포( $p_z(z)$ )를 학습을 통해 비슷하게 만드는 것이 목적
  - label을 통한 분류가 아닌 training data의 분포를 학습
- 확률 분포가 정확히 일치하면 real data와 fake data를 구분할 수 없음

# Generative Adversarial Network

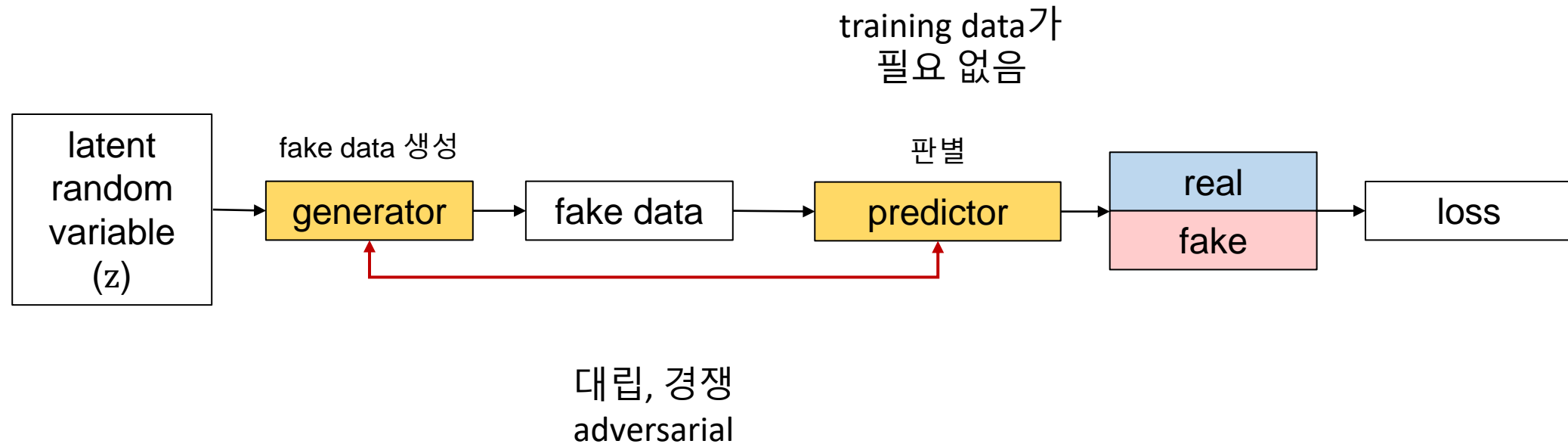
- generator / discriminator



1. **generator**는 discriminator를 속이기 위해 **진짜같은 가짜**를 생성
2. **discriminator**는 generator의 **가짜** 출력을 **판별**하기 위해 학습

# Generative Adversarial Network

- generator / predictor

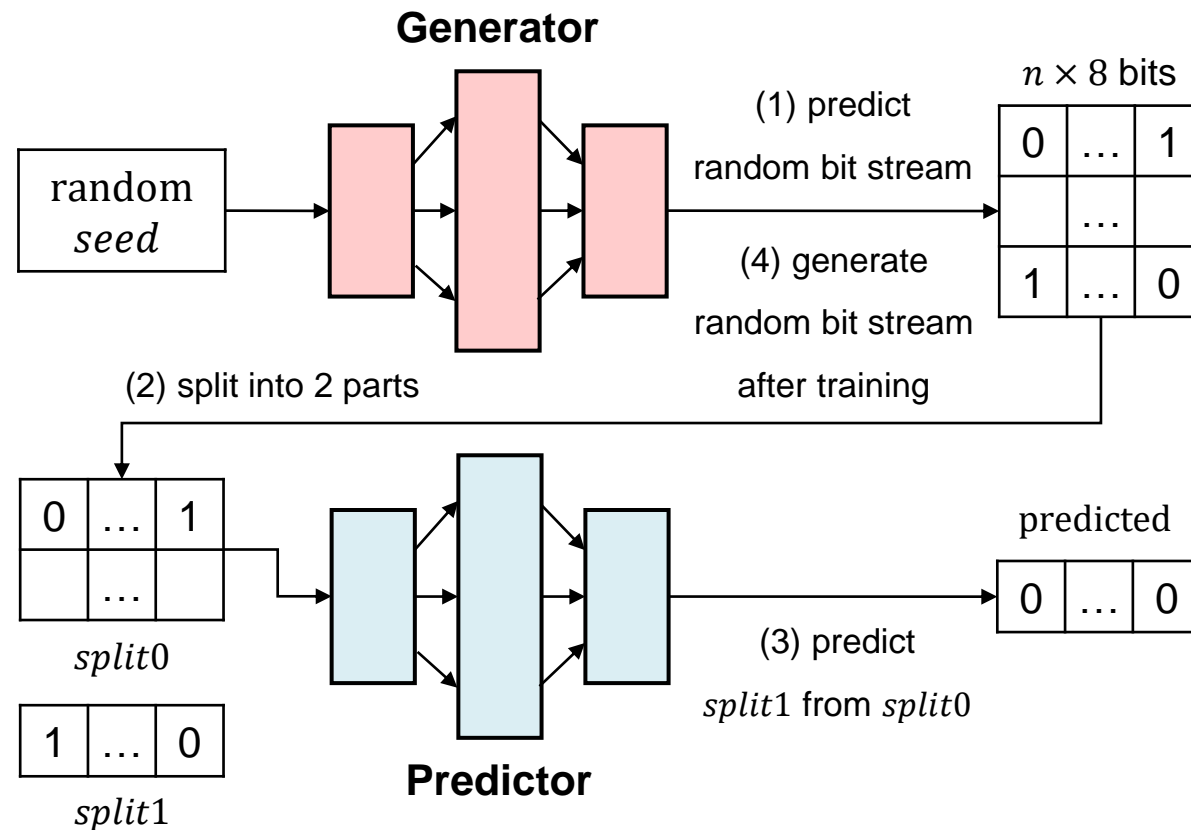


1. **generator**는 predictor 를 속이기 위해 **예측할 수 없는 가짜**를 생성
2. **predictor**는 generator의 가짜데이터를 예측하기 위해 학습

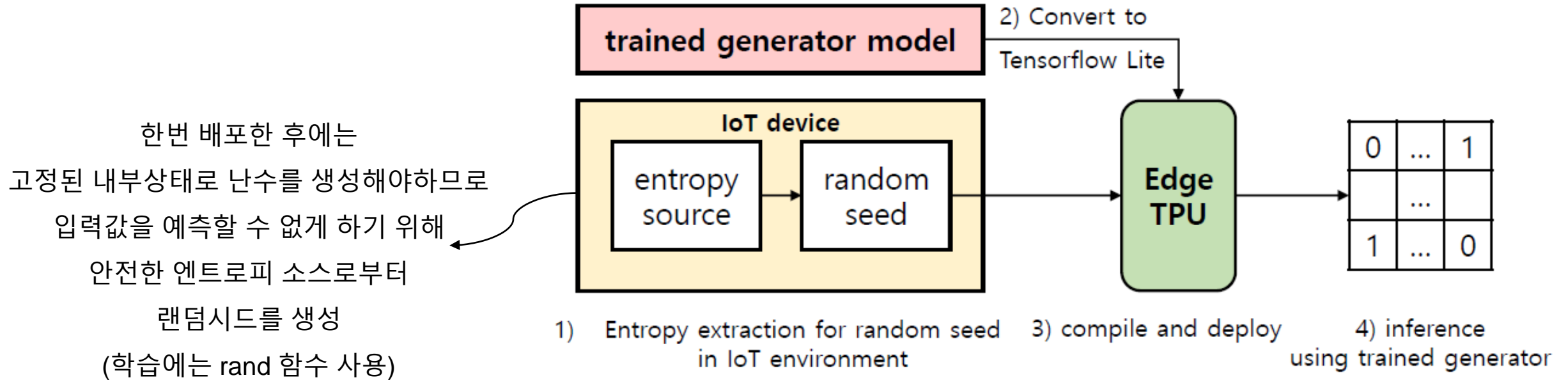
# Proposed Method



# system configuration (1)



## system configuration (2)



random bit stream을 직접 생성하는 것은 generator 이므로  
해당 모델만 edge TPU에 배포



# Comparison with previous method

0 ~ 65535 범위의 10진수 8개 (128bits)

|       |     |   |     |     |
|-------|-----|---|-----|-----|
| 65000 | 381 | 1 | ... | 654 |
| ...   |     |   |     |     |
| 127   |     |   |     |     |

2048 개



- 한번의 batch당 262,144 bits 학습
- 400번 → 104,857,600 bits 생성
- 64bits로 262,144 bits 생성

앞의 십진수 7개로 (112bits) 뒤의 십진수 1개(16bits)를 예측하며 학습

0 or 1 8개 (8bits)

|     |   |   |     |   |
|-----|---|---|-----|---|
| 0   | 1 | 1 | ... | 0 |
| ... |   |   |     |   |
| 1   |   |   |     |   |

137,400 개



- 한번의 batch당 1,099,200 bits 학습
- 100번 → 109,920,000 bits 생성
- 64bits로 1,099,200 bits 생성

앞부분 1,099,192 bits로 뒷부분의 8 bits를 예측하며 학습

# generator

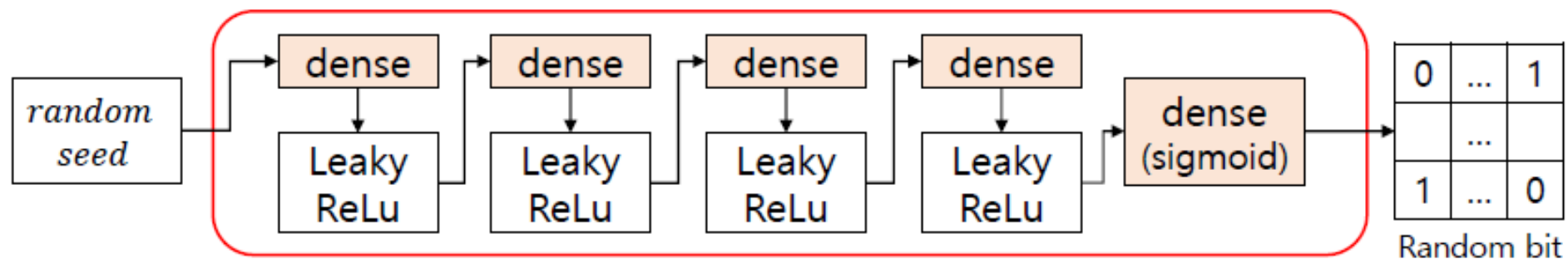


Fig. 3: Architecture of random number generator.

- 가장 기본적인 layer인 dense layer 5층 사용
- 활성화 함수 : LeakyReLU, sigmoid
- 최적화 함수 : Adam (학습률 : 0.0002)

# generator

---

## Algorithm 1 Generator mechanism

---

**Input:** Random seed ( $s$ ), Generator ( $G$ )

**Output:** Random bit stream ( $RBS$ )

```
1:  $x \leftarrow Dense(s)$   
2: for  $i = 1$  to 4 do  
3:    $x \leftarrow Dense(x)$   
4: end for
```

```
5:  $x \leftarrow Sigmoid(x)$ 
```

```
6:  $RBS \leftarrow \text{round } x \text{ into nearest integer (0 or 1)}$ 
```

0~1 사이의 값을 출력 → 0 or 1로 반올림하여 bit로 사용

```
7: return  $RBS$ 
```

---

# predictor

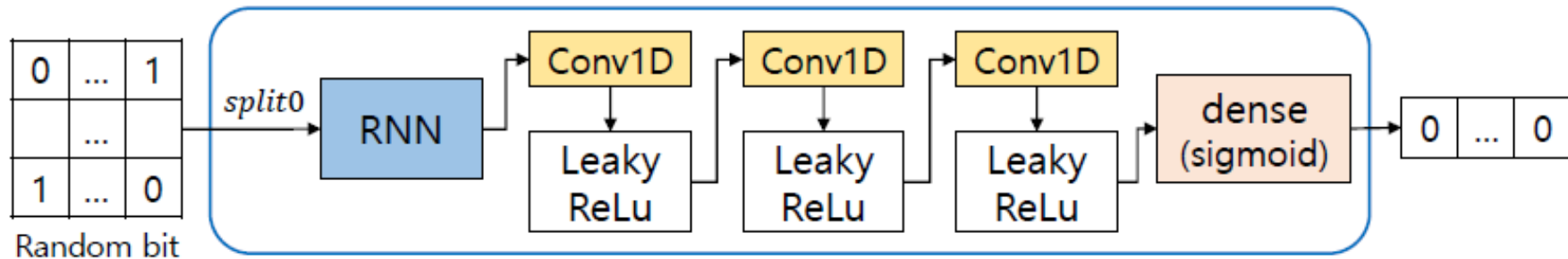


Fig. 4: Architecture of predictor.

- 시퀀스 데이터 학습 layer인 **RNN** 사용, 학습할 가중치가 적으며 성능이 좋은 **Conv1D** layer 사용
- 활성화 함수 : LeakyReLU, sigmoid
- 최적화 함수 : Adam (학습률 : 0.0002)
- **RNN**을 predictor에 넣어 predictor의 성능을 향상시키며, 이에 따라 generator도 학습되도록 함
  - edge TPU에 올리기 위해 generator에는 추가하지 않음  
(올라갈 수 있는 레이어가 제한적이며, 용량이 커질 수 있어서)

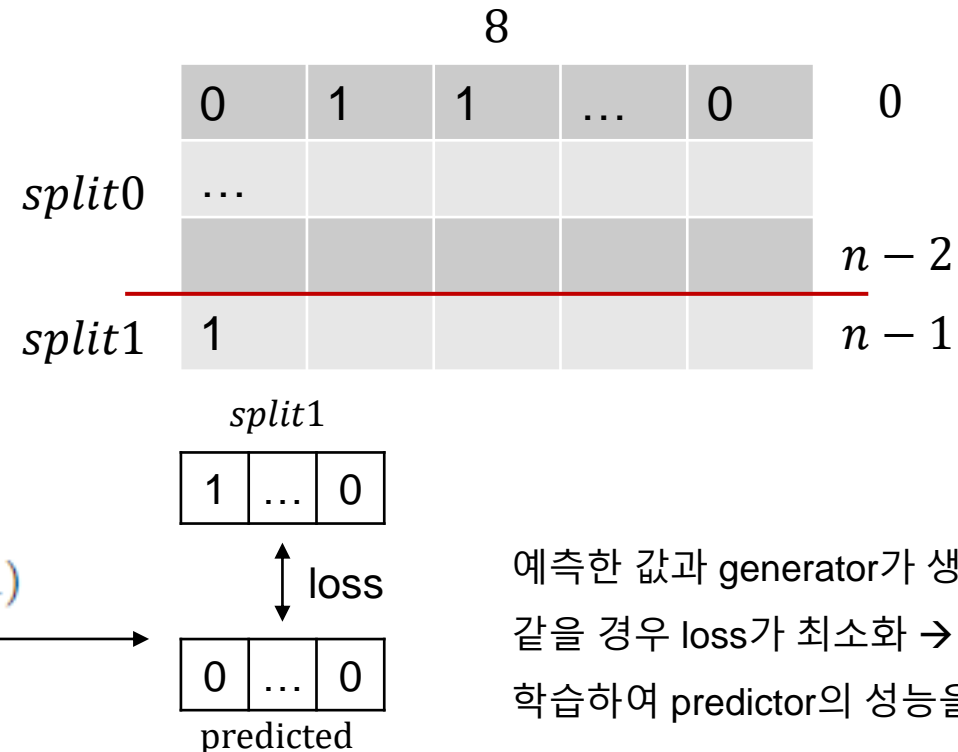
# predictor

## Algorithm 2 Predictor mechanism

**Input:** Random bit stream ( $RBS$ )

**Output:** Predicted random bit stream ( $RBS_P$ )

```
1:  $Split0 \leftarrow RBS[: n - 1][: 8]$ 
2:  $Split1 \leftarrow RBS[n - 1 : n][: 8]$ 
3:  $x \leftarrow RNN(Split0)$ 
4: for  $i = 1$ , to 3 do
5:    $x \leftarrow Conv1D(x)$ 
6: end for
7:  $x \leftarrow Dense(x)$ 
8:  $x \leftarrow Sigmoid(x)$ 
9:  $RBS_P \leftarrow \text{round } x \text{ into nearest integer (0 or 1)}$ 
10:  $Loss_p \leftarrow \text{mean}(\text{abs}(Split1 - RBS_P))$ 
11: Train to minimize  $Loss_p$ 
12: return  $RBS_P, Split1$ 
```



예측한 값과 generator가 생성했던 값이  
같을 경우 loss가 최소화 → 최소화 되도록  
학습하여 predictor의 성능을 향상

# GANtraining (generator training)

## Algorithm 3 Proposed RNG based on GAN

**Input:** Random seed ( $s$ ), Generator ( $G$ ), Predictor ( $P$ ), epochs ( $EPOCHS$ ), Secure parameter ( $t$ ), Range of random number ( $r$ ), The number of bits needed to represent random number ( $m$ )

**Output:** Random Number ( $num$ )

```

1: for epoch = 1 to EPOCHS do
2:    $s \leftarrow$  sample entropy from IoT device
3:    $RBS \leftarrow G(s)$ 
4:    $RBS_P, Split1 \leftarrow P(RBS)$ 
5:    $Loss_G \leftarrow mean(abs(1 - Split1 - RBS_P)) \cdot 0.5$ 
6:   Train  $G$  to minimize  $Loss_G$ 
7:    $RBS \leftarrow G(s)$ 
8: end for
9:  $c \leftarrow \sum_{i=0}^{m+t-1} 2^i \cdot RBS_i$ 
10:  $num \leftarrow c \bmod r$ 
11: return  $num$ 

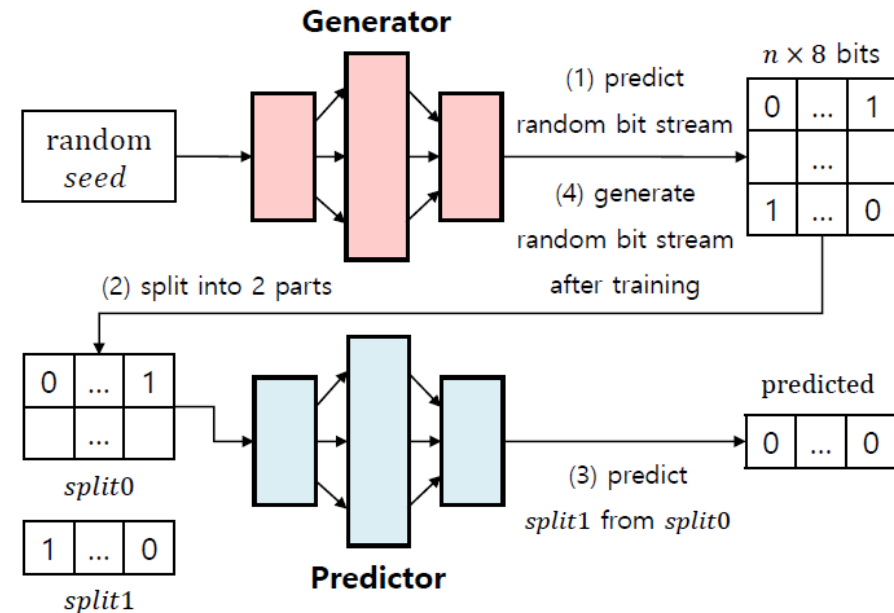
```

$RBS$  to  $num$ 에 필요한 parameter

예측한 값과 generator가 생성한 값이

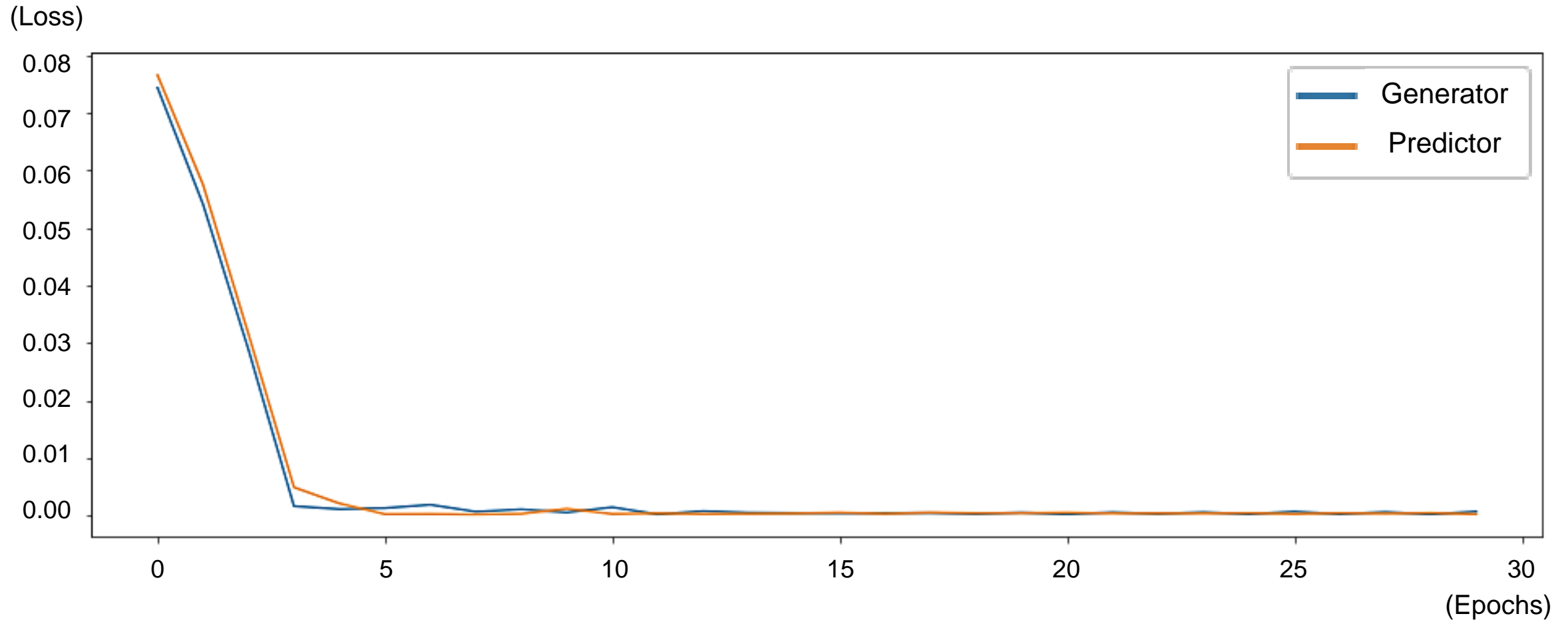
같은 경우 ½, 다른 경우 0

→ 즉 generator는 loss가 최소화 되도록 (predictor가 예측할 수 없도록 학습)



- The Simple Discard Method  
→ 사용되는 비트가 생성 가능한 난수의 최댓값을 표현하기 위한 비트의 ½이하인 경우 비효율
- The Complex Discard Method  
→ 범위를 나눠야해서 복잡함
- The Simple Modular Method  
→ 조건에 따른 반복문이 필요 없어 constant time안에 연산 가능

# GANtraining



loss가 감소하는 형태로 학습된 것을 확인

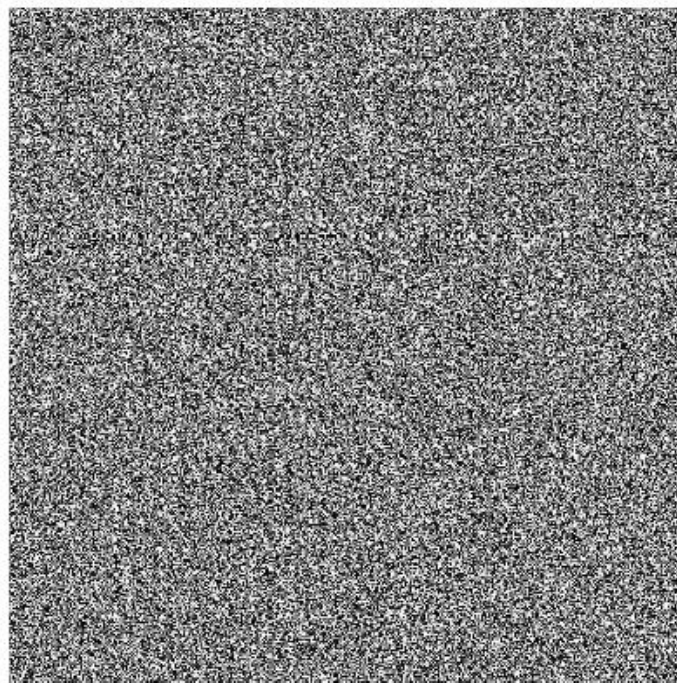
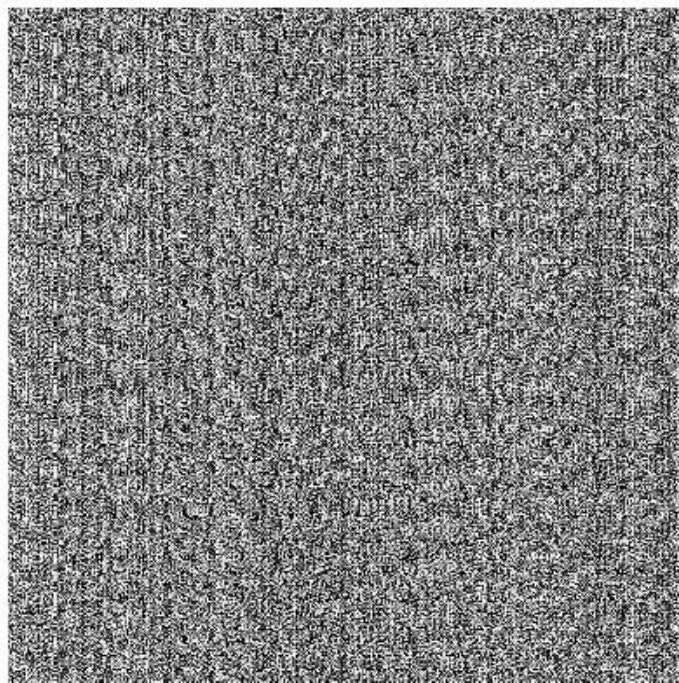
# Evaluation





# Visualization of random number

학습되지 않은 경우  
rand()함수 통해 입력받고  
그것에 대한 출력,  
내부상태 변화 없이  
입력이 반복되므로  
패턴 존재



학습 된 경우,  
내부상태가 변하며  
난수를 생성하도록 학습,  
특정 패턴 없이 분포

Fig. 7: Visualization of random number generated by the generator. (left) before training and (right) after training.

# NST test suite result

generator is <data/1.pi>

| C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 | C10 | P-VALUE | PROPORTION | STATISTICAL TEST |
|----|----|----|----|----|----|----|----|----|-----|---------|------------|------------------|
|----|----|----|----|----|----|----|----|----|-----|---------|------------|------------------|

|   |   |   |   |   |   |   |   |   |   |          |       |                        |
|---|---|---|---|---|---|---|---|---|---|----------|-------|------------------------|
| 0 | 2 | 0 | 0 | 3 | 0 | 2 | 0 | 1 | 2 | 0.213309 | 10/10 | Frequency              |
| 0 | 1 | 0 | 3 | 0 | 1 | 1 | 1 | 2 | 1 | 0.534146 | 10/10 | BlockFrequency         |
| 0 | 1 | 1 | 2 | 1 | 0 | 1 | 2 | 2 | 0 | 0.739918 | 10/10 | CumulativeSums         |
| 1 | 1 | 1 | 1 | 0 | 1 | 2 | 2 | 1 | 0 | 0.911413 | 10/10 | CumulativeSums         |
| 1 | 0 | 0 | 5 | 0 | 0 | 1 | 1 | 0 | 2 | 0.008879 | 10/10 | Runs                   |
| 1 | 1 | 0 | 1 | 2 | 1 | 0 | 1 | 2 | 1 | 0.911413 | 10/10 | LongestRun             |
| 0 | 1 | 1 | 1 | 1 | 2 | 1 | 0 | 0 | 3 | 0.534146 | 10/10 | Rank                   |
| 2 | 3 | 1 | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0.350485 | 10/10 | FFT                    |
| 1 | 1 | 1 | 0 | 1 | 3 | 0 | 1 | 1 | 1 | 0.739918 | 10/10 | NonOverlappingTemplate |
| 0 | 0 | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 0 | 0.213309 | 10/10 | OverlappingTemplate    |
| 1 | 0 | 0 | 0 | 1 | 2 | 1 | 0 | 3 | 2 | 0.350485 | 9/10  | Universal              |
| 0 | 1 | 1 | 1 | 0 | 3 | 0 | 0 | 4 | 0 | 0.035174 | 10/10 | ApproximateEntropy     |
| 0 | 0 | 0 | 2 | 0 | 1 | 2 | 0 | 0 | 2 | -----    | 7/7   | RandomExcursions       |
| 2 | 0 | 1 | 3 | 0 | 0 | 1 | 0 | 1 | 2 | 0.350485 | 10/10 | Serial                 |
| 1 | 1 | 1 | 0 | 1 | 2 | 0 | 0 | 1 | 3 | 0.534146 | 10/10 | Serial                 |
| 0 | 1 | 0 | 3 | 2 | 0 | 2 | 2 | 0 | 0 | 0.213309 | 10/10 | LinearComplexity       |

15 개의 test  
188개의 Individual test  
NonOverlappingTemplate은  
여러 번 반복



# NST test suite result

| generator is <data/1.pi> |    |    |    |    |                          |    |    |    |     |          |       | generator is <data/5.pi> |    |    |    |    |    |    |    |    |          |          |                | generator is <data/9.pi> |    |    |    |    |    |    |    |    |    |          |          |                |                         |                         |   |   |   |          |       |      |  |  |  |  |  |
|--------------------------|----|----|----|----|--------------------------|----|----|----|-----|----------|-------|--------------------------|----|----|----|----|----|----|----|----|----------|----------|----------------|--------------------------|----|----|----|----|----|----|----|----|----|----------|----------|----------------|-------------------------|-------------------------|---|---|---|----------|-------|------|--|--|--|--|--|
| C1                       | C2 | C3 | C4 | C5 | C6                       | C7 | C8 | C9 | C10 | P-VALUE  |       | C1                       | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 | C10      | P-VALUE  | PROPORTION     | STAT                     | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 | C10      | P-VALUE  | PROPORTION     | STATISTICAL TEST        |                         |   |   |   |          |       |      |  |  |  |  |  |
| 0                        | 2  | 0  | 0  | 3  | 0                        | 2  | 0  | 1  | 2   | 0.213309 | 10/10 | 1                        | 0  | 1  | 2  | 2  | 0  | 1  | 1  | 0  | 2        | 0.739918 | 10/10          | Frequency                | 0  | 1  | 3  | 1  | 1  | 1  | 2  | 0  | 0  | 1        | 0.534146 | 10/10          | Frequency               |                         |   |   |   |          |       |      |  |  |  |  |  |
| 0                        | 1  | 0  | 3  | 0  | 1                        | 1  | 1  | 2  | 1   | 0.534146 | 10/10 | 1                        | 0  | 1  | 1  | 1  | 1  | 2  | 1  | 1  | 0.991468 | 10/10    | BlockFrequency | 1                        | 0  | 2  | 1  | 1  | 1  | 1  | 0  | 1  | 2  | 0.911413 | 10/10    | BlockFrequency |                         |                         |   |   |   |          |       |      |  |  |  |  |  |
| 0                        | 1  | 1  | 2  | 1  | 0                        | 1  | 2  | 2  | 0   | 0.739918 | 10/10 | 1                        | 0  | 1  | 2  | 3  | 1  | 1  | 0  | 0  | 1        | 0.534146 | 10/10          | CumulativeSums           | 0  | 1  | 0  | 1  | 3  | 2  | 2  | 1  | 0  | 0        | 0.350485 | 10/10          | CumulativeSums          |                         |   |   |   |          |       |      |  |  |  |  |  |
| 1                        | 1  | 1  | 1  | 0  | 1                        | 2  | 2  | 1  | 0   | 0.911413 | 10/10 | 1                        | 0  | 1  | 2  | 2  | 1  | 0  | 1  | 1  | 1        | 0.911413 | 10/10          | CumulativeSums           | 2  | 1  | 0  | 1  | 1  | 0  | 1  | 2  | 2  | 0        | 0.739918 | 10/10          | CumulativeSums          |                         |   |   |   |          |       |      |  |  |  |  |  |
| 1                        | 0  | 0  | 5  | 0  | generator is <data/3.pi> |    |    |    |     |          |       | 0                        | 1  | 3  | 0  | 3  | 0  | 0  | 1  | 1  | 1        | 0.213309 | 10/10          | Runs                     | 3  | 0  | 1  | 1  | 1  | 0  | 1  | 0  | 2  | 1        | 0.534146 | 10/10          | Runs                    |                         |   |   |   |          |       |      |  |  |  |  |  |
| 1                        | 1  | 0  | 1  | 2  |                          |    |    |    |     |          |       | 0                        | 0  | 1  | 3  | 1  | 1  | 2  | 1  | 0  | 1        | 0.534146 |                | generator is <data/7.pi> | 0  | 1  | 2  | 0  | 2  | 1  | 0  | 3  | 0  | 1        | 0.350485 | 10/10          | LongestRun              |                         |   |   |   |          |       |      |  |  |  |  |  |
| 0                        | 1  | 1  | 1  | 1  | C1                       | C2 | C3 | C4 | C5  | C6       | C7    | C8                       | 0  | 1  | 2  | 0  | 0  | 4  | 0  | 2  | 0        | 1        | 0.066882       |                          | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | 0  | 1        | 1        | 0              | 1                       | 1                       | 0 | 3 | 2 | 0.534146 | 10/10 | Rank |  |  |  |  |  |
| 2                        | 3  | 1  | 2  | 1  |                          |    |    |    |     |          |       | 1                        | 2  | 0  | 2  | 0  | 1  | 1  | 1  | 1  | 1        | 0.911413 |                |                          | 2  | 0  | 1  | 0  | 1  | 0  | 2  | 1  | 1  | 2        | 0.739918 | 10/10          | FFT                     |                         |   |   |   |          |       |      |  |  |  |  |  |
| 0                        | 0  | 1  | 1  | 1  |                          |    |    |    |     |          |       | 0                        | 1  | 1  | 0  | 2  | 2  | 0  | 1  | 2  | 1        | 0.739918 |                |                          | 0  | 0  | 0  | 2  | 0  | 0  | 0  | 1  | 4  | 3        | 0.017912 | 10/10          | NonOverlappingTemplate  |                         |   |   |   |          |       |      |  |  |  |  |  |
| 1                        | 0  | 0  | 0  | 1  |                          |    |    |    |     |          |       | 1                        | 0  | 1  | 1  | 2  | 3  | 0  | 1  | 1  | 0        | 0.534146 |                |                          | 0  | 1  | 1  | 2  | 0  | 0  | 1  | 0  | 3  | 2        | 0.350485 | 10/10          | Universal               |                         |   |   |   |          |       |      |  |  |  |  |  |
| 0                        | 1  | 1  | 1  | 0  |                          |    |    |    |     |          |       | 4                        | 2  | 1  | 0  | 1  | 0  | 1  | 1  | 0  | 0        | 0.122325 |                |                          | 2  | 2  | 0  | 1  | 0  | 0  | 2  | 0  | 1  | 2        | 0.534146 | 10/10          | ApproximateEntropy      |                         |   |   |   |          |       |      |  |  |  |  |  |
| 0                        | 0  | 0  | 2  | 0  |                          |    |    |    |     |          |       | 1                        | 2  | 0  | 0  | 2  | 0  | 0  | 2  | 1  | 2        | 0.534146 |                |                          | 0  | 2  | 0  | 1  | 1  | 2  | 0  | 2  | 0  | 2        |          |                | 6/6                     | RandomExcursions        |   |   |   |          |       |      |  |  |  |  |  |
| 0                        | 0  | 0  | 1  | 1  |                          |    |    |    |     |          |       | 0                        | 2  | 0  | 1  | 0  | 0  | 0  | 1  | 1  | 0        | ----     |                |                          | 0  | 0  | 2  | 0  | 1  | 1  | 2  | 1  | 2  | 1        |          |                | 6/6                     | RandomExcursionsVariant |   |   |   |          |       |      |  |  |  |  |  |
| 2                        | 0  | 1  | 3  | 0  |                          |    |    |    |     |          |       | 0                        | 0  | 1  | 0  | 0  | 0  | 2  | 0  | 0  | 2        | ----     |                |                          | 1  | 0  | 0  | 1  | 1  | 1  | 2  | 1  | 2  | 1        |          |                |                         |                         |   |   |   |          |       |      |  |  |  |  |  |
| 1                        | 1  | 1  | 0  | 1  |                          |    |    |    |     |          |       | 1                        | 0  | 1  | 0  | 2  | 2  | 1  | 2  | 0  | 1        | 0.739918 |                |                          | 1  | 0  | 0  | 1  | 1  | 0  | 2  | 2  | 1  | 2        |          |                |                         |                         |   |   |   |          |       |      |  |  |  |  |  |
| 0                        | 1  | 0  | 3  | 2  |                          |    |    |    |     |          |       | 0                        | 3  | 2  | 0  | 0  | 1  | 2  | 0  | 0  | 2        | 0.213309 |                |                          | 1  | 2  | 1  | 0  | 0  | 3  | 0  | 1  | 0  | 2        |          |                |                         |                         |   |   |   |          |       |      |  |  |  |  |  |
|                          |    |    |    |    |                          |    |    |    |     |          |       | 1                        | 3  | 1  | 0  | 0  | 1  | 0  | 3  | 1  | 0        | 0.213309 |                |                          | 0  | 2  | 0  | 1  | 0  | 4  | 2  | 1  | 0  | 0        |          |                |                         |                         |   |   |   |          |       |      |  |  |  |  |  |
|                          |    |    |    |    |                          |    |    |    |     |          |       | 1                        | 0  | 0  | 1  | 1  | 1  | 0  | 2  | 0  | 4        | 0.122325 | 10/10          | NonOverlappingTemplate   | 1  | 0  | 3  | 2  | 1  | 0  | 1  | 0  | 0  | 2        | 0.350485 | 10/10          | NonOverlappingTemplate  |                         |   |   |   |          |       |      |  |  |  |  |  |
|                          |    |    |    |    |                          |    |    |    |     |          |       | 0                        | 1  | 0  | 3  | 1  | 1  | 1  | 1  | 1  | 1        | 0.739918 | 10/10          | OverlappingTemplate      | 2  | 0  | 1  | 3  | 2  | 1  | 0  | 0  | 1  | 0        | 0.350485 | 10/10          | OverlappingTemplate     |                         |   |   |   |          |       |      |  |  |  |  |  |
|                          |    |    |    |    |                          |    |    |    |     |          |       | 0                        | 0  | 1  | 3  | 3  | 1  | 2  | 0  | 0  | 0        | 0.122325 | 10/10          | Universal                | 0  | 0  | 2  | 0  | 1  | 0  | 3  | 0  | 0  | 4        | 0.017912 | 10/10          | Universal               |                         |   |   |   |          |       |      |  |  |  |  |  |
|                          |    |    |    |    |                          |    |    |    |     |          |       | 0                        | 2  | 1  | 0  | 3  | 0  | 1  | 1  | 2  | 0        | 0.350485 | 10/10          | ApproximateEntropy       | 0  | 0  | 2  | 1  | 0  | 3  | 1  | 2  | 1  | 0        | 0.350485 | 10/10          | ApproximateEntropy      |                         |   |   |   |          |       |      |  |  |  |  |  |
|                          |    |    |    |    |                          |    |    |    |     |          |       | 0                        | 0  | 2  | 0  | 1  | 0  | 2  | 0  | 1  | 0        | ----     | 6/6            | RandomExcursions         | 2  | 1  | 1  | 0  | 2  | 1  | 2  | 0  | 0  | 0        | ----     | 9/9            | RandomExcursions        |                         |   |   |   |          |       |      |  |  |  |  |  |
|                          |    |    |    |    |                          |    |    |    |     |          |       | 0                        | 0  | 1  | 1  | 1  | 0  | 0  | 0  | 2  | 1        | ----     | 6/6            | RandomExcursionsVariant  | 1  | 0  | 1  | 2  | 1  | 2  | 0  | 0  | 0  | 2        | ----     | 9/9            | RandomExcursionsVariant |                         |   |   |   |          |       |      |  |  |  |  |  |
|                          |    |    |    |    |                          |    |    |    |     |          |       | 0                        | 1  | 0  | 1  | 1  | 2  | 0  | 1  | 2  | 2        | 0.739918 | 10/10          | Serial                   | 1  | 2  | 0  | 3  | 0  | 1  | 1  | 0  | 1  | 1        | 0.534146 | 9/10           | Serial                  |                         |   |   |   |          |       |      |  |  |  |  |  |
|                          |    |    |    |    |                          |    |    |    |     |          |       | 1                        | 1  | 1  | 0  | 0  | 2  | 2  | 0  | 1  | 2        | 0.739918 | 9/10           | Serial                   | 1  | 1  | 0  | 1  | 1  | 2  | 1  | 2  | 1  | 0        | 0.911413 | 9/10           | Serial                  |                         |   |   |   |          |       |      |  |  |  |  |  |
|                          |    |    |    |    |                          |    |    |    |     |          |       | 0                        | 2  | 0  | 1  | 0  | 0  | 2  | 0  | 3  | 2        | 0.213309 | 10/10          | LinearComplexity         | 1  | 0  | 4  | 0  | 0  | 1  | 1  | 1  | 1  | 1        | 0.213309 | 9/10           | LinearComplexity        |                         |   |   |   |          |       |      |  |  |  |  |  |
|                          |    |    |    |    |                          |    |    |    |     |          |       | 2                        | 0  | 1  | 2  | 3  | 0  | 1  | 0  | 1  | 0        | 0.350485 | 9/10           | LinearComplexity         | 2  | 0  | 1  | 2  | 3  | 0  | 1  | 0  | 1  | 0        | 0.350485 | 9/10           | LinearComplexity        |                         |   |   |   |          |       |      |  |  |  |  |  |

# NST test suite result

- The following test instances were improved

| generator is <data/c-1.pi> |    |    |    |    |    |    |    |    |     |            |            |                         |  |
|----------------------------|----|----|----|----|----|----|----|----|-----|------------|------------|-------------------------|--|
| C1                         | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 | C10 | P-VALUE    | PROPORTION | STATISTICAL TEST        |  |
| 10                         | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0.000000 * | 0/10       | * Frequency             |  |
| 2                          | 2  | 3  | 0  | 0  | 1  | 1  | 1  | 0  | 0   | 0.350485   | 9/10       | BlockFrequency          |  |
| 10                         | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0.000000 * | 0/10       | * CumulativeSums        |  |
| 10                         | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0.000000 * | 0/10       | * CumulativeSums        |  |
| 9                          | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0   | 0.000000 * | 1/10       | * Runs                  |  |
| 2                          | 1  | 0  | 2  | 1  | 0  | 1  | 0  | 3  | 0   | 0.350485   | 10/10      | LongestRun              |  |
| 0                          | 0  | 1  | 5  | 2  | 2  | 0  | 0  | 0  | 0   | 0.004301   | 10/10      | Rank                    |  |
| 4                          | 4  | 1  | 0  | 0  | 0  | 0  | 0  | 1  | 0   | 0.004301   | 8/10       | FFT                     |  |
| 3                          | 0  | 1  | 1  | 2  | 0  | 0  | 2  | 0  | 1   | 0.350485   | 8/10       | NonOverlappingTemplate  |  |
| 3                          | 0  | 1  | 1  | 2  | 1  | 1  | 0  | 1  | 0   | 0.534146   | 10/10      | OverlappingTemplate     |  |
| 0                          | 0  | 1  | 1  | 2  | 2  | 1  | 3  | 0  | 0   | 0.350485   | 10/10      | Universal               |  |
| 3                          | 4  | 0  | 0  | 2  | 1  | 0  | 0  | 0  | 0   | 0.017912   | 9/10       | ApproximateEntropy      |  |
| 0                          | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | ----       | 7/7        | RandomExcursions        |  |
| 0                          | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | ----       | 7/7        | RandomExcursionsVariant |  |
| 0                          | 1  | 3  | 1  | 1  | 1  | 0  | 0  | 1  | 2   | 0.534146   | 10/10      | Serial                  |  |
| 1                          | 3  | 0  | 0  | 1  | 1  | 1  | 0  | 2  | 1   | 0.534146   | 10/10      | Serial                  |  |
| 1                          | 1  | 1  | 1  | 2  | 0  | 0  | 1  | 1  | 2   | 0.911413   | 10/10      | LinearComplexity        |  |

previous method

| generator is <data/1.pi> |    |    |    |    |    |    |    |    |     |          |            |                         |  |
|--------------------------|----|----|----|----|----|----|----|----|-----|----------|------------|-------------------------|--|
| C1                       | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 | C10 | P-VALUE  | PROPORTION | STATISTICAL TEST        |  |
| 0                        | 2  | 0  | 0  | 3  | 0  | 2  | 0  | 1  | 2   | 0.213309 | 10/10      | Frequency               |  |
| 0                        | 1  | 0  | 3  | 0  | 1  | 1  | 1  | 2  | 1   | 0.534146 | 10/10      | BlockFrequency          |  |
| 0                        | 1  | 1  | 2  | 1  | 0  | 1  | 2  | 2  | 0   | 0.739918 | 10/10      | CumulativeSums          |  |
| 1                        | 1  | 1  | 1  | 0  | 1  | 2  | 2  | 1  | 0   | 0.911413 | 10/10      | CumulativeSums          |  |
| 1                        | 0  | 0  | 5  | 0  | 0  | 1  | 1  | 0  | 2   | 0.008879 | 10/10      | Runs                    |  |
| 1                        | 1  | 0  | 1  | 2  | 1  | 0  | 1  | 2  | 1   | 0.911413 | 10/10      | LongestRun              |  |
| 0                        | 1  | 1  | 1  | 1  | 2  | 1  | 0  | 0  | 3   | 0.534146 | 10/10      | Rank                    |  |
| 2                        | 3  | 1  | 2  | 1  | 1  | 0  | 0  | 0  | 0   | 0.350485 | 10/10      | FFT                     |  |
| 1                        | 1  | 1  | 0  | 1  | 3  | 0  | 1  | 1  | 1   | 0.739918 | 10/10      | NonOverlappingTemplate  |  |
| 0                        | 0  | 1  | 1  | 1  | 1  | 4  | 1  | 1  | 0   | 0.213309 | 10/10      | OverlappingTemplate     |  |
| 1                        | 0  | 0  | 0  | 1  | 2  | 1  | 0  | 3  | 2   | 0.350485 | 9/10       | Universal               |  |
| 0                        | 1  | 1  | 1  | 0  | 3  | 0  | 0  | 4  | 0   | 0.035174 | 10/10      | ApproximateEntropy      |  |
| 0                        | 0  | 0  | 2  | 0  | 1  | 2  | 0  | 0  | 2   | ----     | 7/7        | RandomExcursions        |  |
| 0                        | 0  | 0  | 1  | 1  | 0  | 0  | 2  | 2  | 1   | ----     | 7/7        | RandomExcursionsVariant |  |
| 2                        | 0  | 1  | 3  | 0  | 0  | 1  | 0  | 1  | 2   | 0.350485 | 10/10      | Serial                  |  |
| 1                        | 1  | 1  | 0  | 1  | 2  | 0  | 0  | 1  | 3   | 0.534146 | 10/10      | Serial                  |  |
| 0                        | 1  | 0  | 3  | 2  | 0  | 2  | 2  | 0  | 0   | 0.213309 | 10/10      | LinearComplexity        |  |

proposed method

# previous work vs this work

Table 1: Comparison of GAN based PRNG, where  $T$ ,  $T_I$ ,  $F_I$ ,  $F_I/\%$ ,  $F_P$ ,  $F_T$ ,  $F\%$  are the number of individual tests, test instances, failed instances, their percentage, individual tests with p-value below the threshold, individual tests that failed, their percentage, respectively. The inference time is the time to generate a random number through trained generator.

|                     | $T$ | $T_I$ | $F_I$ | $F_I/\%$ | $F_P$ | $F_T$ | $F\%$ | inference time |
|---------------------|-----|-------|-------|----------|-------|-------|-------|----------------|
| Before training     | 188 | 1789  | 1769  | 98.8     | 160.8 | 186   | 98.9  | 177.32 ms      |
| Bernardi et al. [5] | 188 | 1830  | 56    | 3.0      | 2.7   | 4.5   | 2.5   | 187.09 ms      |
| Proposed method     | 188 | 1794  | 19.6  | 1.09     | 0.00  | 0.1   | 0.00  | 13.27 ms       |

↓  
약 2.85배 감소

↓  
약 45배 감소

↓  
약 14.1배 향상  
egde TPU가 보통  
15~30배 빠르다고 함

|   |   |   |   |   |   |   |   |   |   |      |     |                  |
|---|---|---|---|---|---|---|---|---|---|------|-----|------------------|
| 0 | 0 | 2 | 0 | 1 | 0 | 2 | 0 | 1 | 0 | ---- | 6/6 | RandomExcursions |
| 0 | 0 | 0 | 2 | 0 | 1 | 2 | 0 | 0 | 2 | ---- | 7/7 | RandomExcursions |

→ 각 실험마다 다른 수치가 나와서  
기존 논문의 결과와 *test instance* ( $T_I$ )가 다름

# previous work vs this work

- ❖ 기존 연구의 경우 통과하지 못 한 test 항목은 주로 Frequency, CumulativeSums, Run, FFT, NonOverlappingTemplate.
- ❖ 제안 기법의 경우 전체 테스트(188개의 개별 테스트에 대한 10번의 실험)에 대해 NonOverlappingTemplate 1번 실패

- **Frequency**

4 0 2 0 0 1 1 2 0 0 0.066882 7/10 \* NonOverlappingTemplate

0과 1의 비율 (이상적인 경우 0.5에 수렴)

- **CumulativeSums**

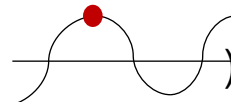
random walk test

0은 -1, 1은 1로 하여 누적합을 계산하며, 이상적인 난수열의 결과 값(0)과의 차이를 검사

- **Run**

같은 비트들이 반복되는 것을 run이라고 하며, 0의 run이 1의 run으로 변하는 것의 확률을 검사 (이상적인 경우 0.5)

- **FFT**

이산 푸리에 변환의 최고점의 높이()를 활용하여 반복적 패턴, 주기 등을 검사

- **NonOverlappingTemplate**

비주기적인 패턴의 빈도수 검사 (특정 m-bits 패턴을 찾음)

# previous work vs this work

## ❖ 기존 연구의 경우

covolution layer만 사용 → 특정 패턴 존재, 생성되는 비트들의 빈도수 등에 있어 이상적인 난수열과 거리가 있음을 확인

## ❖ RNN 레이어를 통해 보다 긴 시퀀스에 대해 학습

- 장기적 의존성을 가짐 → 이전 bit들의 값과 전체적 특징을 반영하여 학습하고 random bit stream 생성
- 기존 논문과 동일하게 각 test instanc당 약 100만 비트를 기준으로 실험
- but 기존 연구의 경우 262,144 bits 씩 학습하였고 제안 기법의 경우 1,099,200 bits씩 학습
- 지역적 특징(convolution layer) 으로 학습 및 생성된 난수열에 비해  
전체적 시퀀스를 학습하여 생성된 난수열이 더 좋은 성능을 나타냄
- NIST test suite 결과를 보아, 제안 기법은 랜덤 시드 생성을 위해 안전한 entropy source를 사용한다면 CSPRNG(Cryptographically Secure PRNG)로 사용 가능할 것

Q & A

