TurboQuant: Online Vector Quantization with Near-optimal Distortion Rate

Dooyoung Kim
Natural Language Processing Lab, SKKU

성균관대학교
SUNG KYUN KWAN UNIVERSITY(SKKU)

Contents

KV Cache is a Memory Bottleneck

LLM의 Inference

Masked Multi-Head Self-Attention

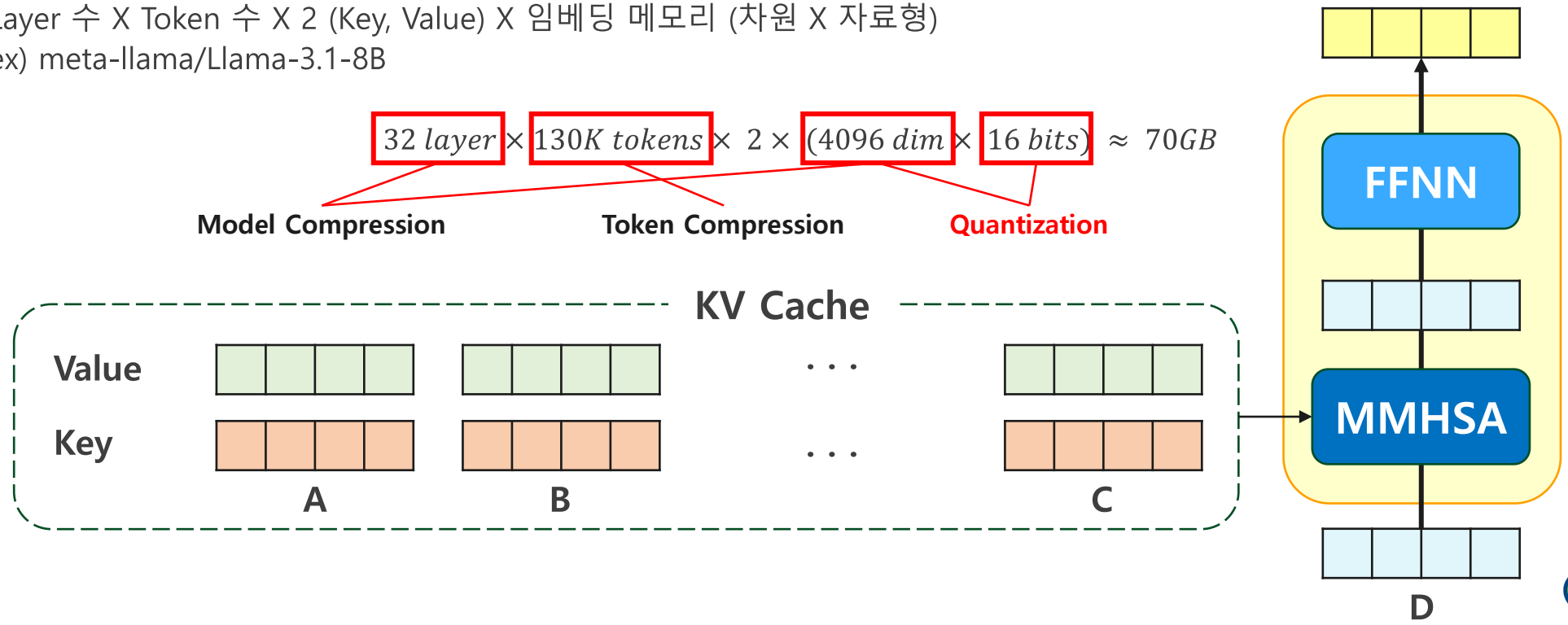
- 기존의 저장된 Key, Value 벡터와 attention 연산 (KV Cache)

입력 토큰이 길어지면 KV Cache가 대부분의 메모리를 차지

- GPT-5 400K tokens, Claude 200K tokens, Llama-3.1-8B 130K tokens
- Layer 수 X Token 수 X 2 (Key, Value) X 임베딩 메모리 (차원 X 자료형)
  ex) meta-llama/Llama-3.1-8B
-

$$32\ layer \times 130K\ tokens \times 2 \times (4096\ dim \times 16\ bits) \approx 70GB$$

Model Compression | Token Compression | Quantization



Information Theory

Entropy & Mutual Information

Entropy $H(x)$

- 확률변수가 가지는 **불확실성의 정도** (평균적으로 얼마나 많은 정보가 필요한가)
- $H(X) = -\sum_x p(x) \log p(x)$

Mutual Information $I(X;Y)$

- 두 확률 변수 X와 Y가 서로에 대해 **얼마나 많은 정보를 공유**하는가 (상호 정보량)
- $I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} = H(X) - H(X|Y)$
- $I(X;Y) = 0$: 두 확률 변수 X와 Y가 완전히 독립

Y / X	A	B	C
α	1/12	1/12	2/12
β	2/12	2/12	4/12

$p(X|\alpha) = p(X|\beta) \rightarrow I(X;Y) = 0$

Y / X	A	B	C
α	2/12	1/12	2/12
β	1/12	2/12	4/12

$p(X|\alpha) \neq p(X|\beta) \rightarrow I(X;Y) > 0$

< 정보 이론 관점, Huffman Coding >

- $A = 0, B = 10, C = 11$
- $ABACCAC \leftrightarrow 0101111011$

$$\mathbb{E}(bits) = 0.4 * 1 + 0.2 * 2 + 0.4 * 2 = 1.6\ bits$$

$$H(X|\alpha) = 1.522 \ldots bits$$

$$\mathbb{E}(bits) \geq Entropy$$

Information Theory

Quantization & Distortion

Quantization

- 더 적은 bit 또는 차원으로 데이터를 표현하는 방법 (연산 or 메모리 효율)
- 필연적으로 정보의 손실이 발생 (Distortion)

Lemma 2. Shannon Lower Bound (SLB, Shannon's lossy source coding theorem)

- 확률 분포 p_X를 따르고 유한한 미분가능 엔트로피 $h(x)$를 갖는 랜덤 벡터 $x \in \mathbb{R}^d$ 에 대해서, B bit로 압축 후 복원한 벡터 y와의 MSE 오차 $D(p_X, B)$ 는 다음과 같다.

$$D(p_X, B) \coloneqq \inf\{\mathbb{E}[\|x - y\|_2^2] : I(x; y) \leq B\}$$

- 이때 $D(p_X, B)$ 는 mutual information $I(x; y)$가 최대 B인 모든 x와 y의 결합분포의 infimum(하한)이며, 다음과 같은 값을 갖는다.

$$D(p_X, B) \geq \frac{d}{2\pi e} 2^{(2/d)(h(x)-B)}$$

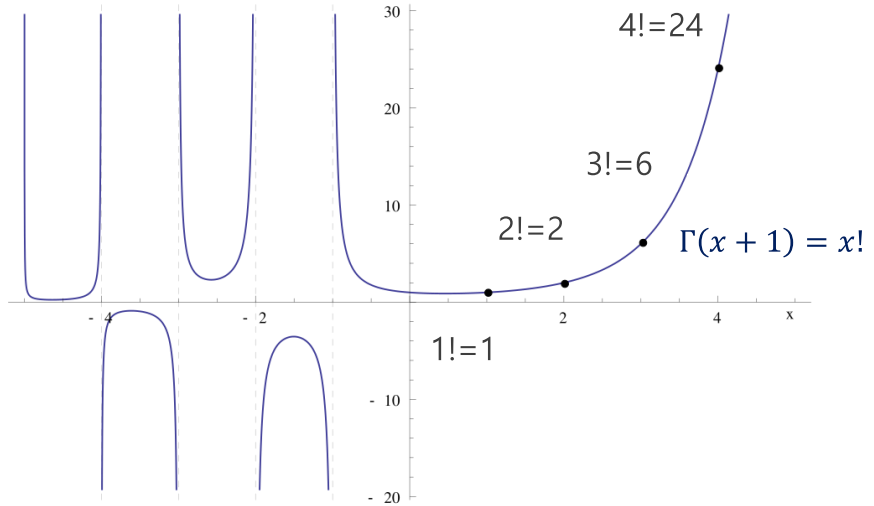
Mathematics

Gamma Function Γ

- **Factorial 함수**를 복소수 범위로 확장한 함수 (음의 정수에 대해서는 정의 X)

$$\Gamma(z) = \int_0^\infty e^{-t} t^{z-1} dt$$

$$\Gamma(n) = (n-1)!$$



- **d차원 Sphere**
  - d 차원 구의 부피와 표면적을 구하는데 Gamma Function을 사용
  - Volume: $V_d(r) = \frac{\pi^{d/2}}{\Gamma(\frac{d}{2}+1)} r^d$
  - Surface: $S_d(r) = \frac{2\pi^{d/2}}{\Gamma(\frac{d}{2})} r^{d-1}$

$\Gamma(1)$	$\Gamma(1.5)$	$\Gamma(2)$
1	$\sqrt{\pi}/2$	1

	d=2	d=3
Volume	πr^2	$\frac{4}{3}\pi r^3$
Surface	$2\pi r$	$4\pi r^2$

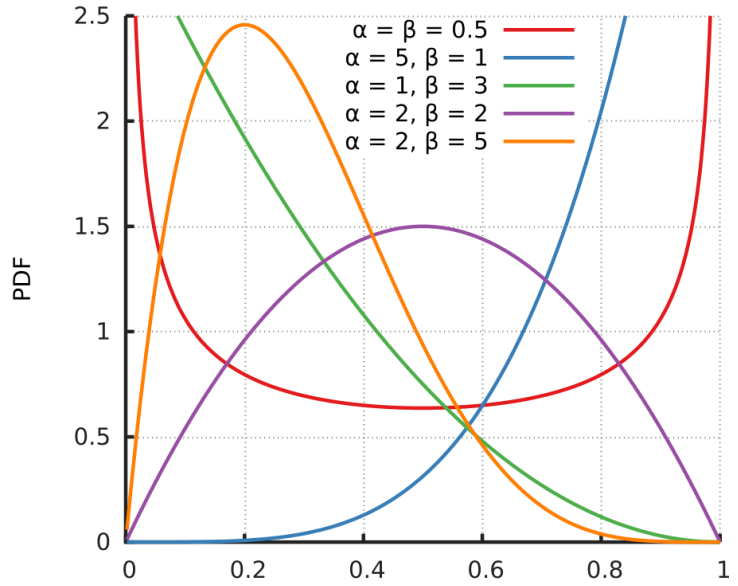
Mathematics

Beta Distribution

- 확률 자체를 확률적으로 모델링하기 위한 함수
- $[0,1]$에서 정의되는 연속 확률 분포

$$f(x;\alpha,\beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)}$$

$$B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$$



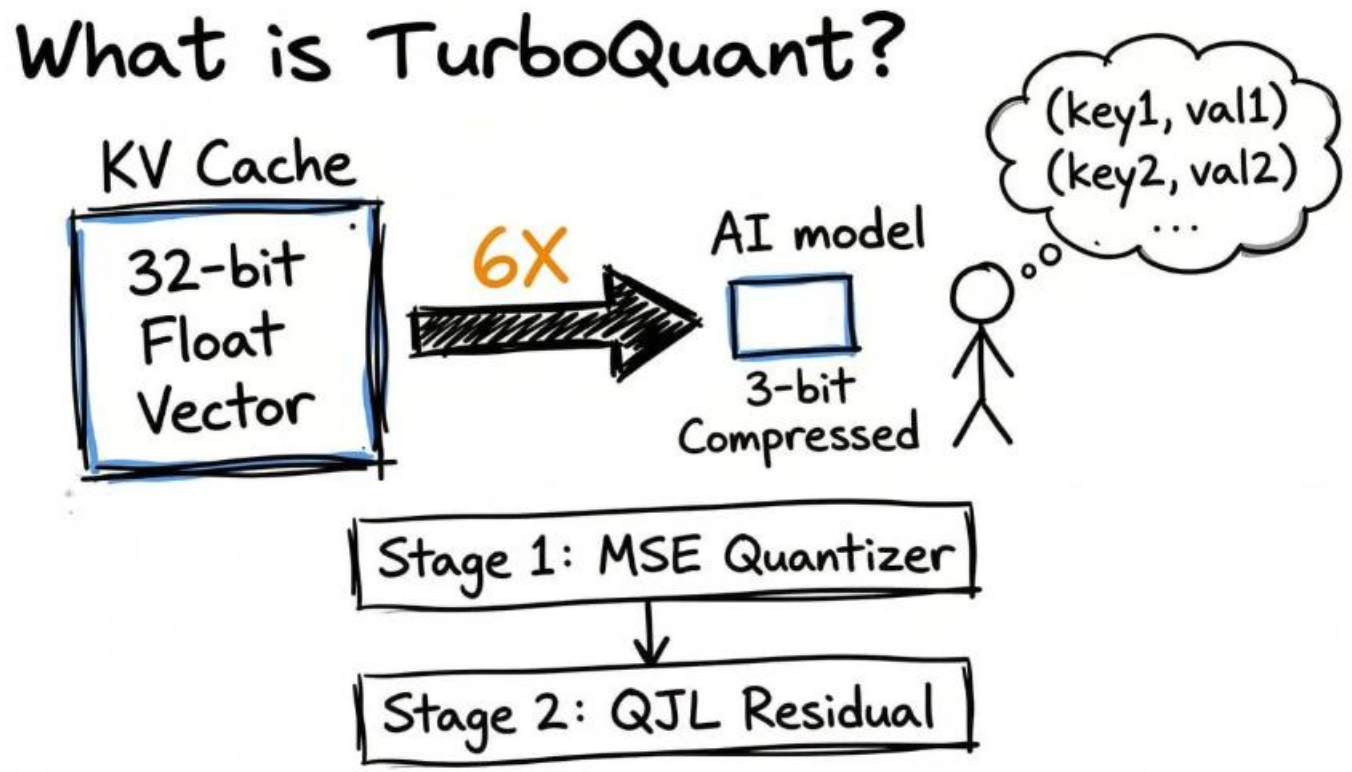
- **주요 특성**
  - $\mathbb{E}(X) = \frac{\alpha}{\alpha+\beta}$
  - $Var(X) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$

앞면 +0 / 뒷면+2 앞면 +1 / 뒷면+2

앞면	1	$1+0=1$	$1+1=2$
뒷면	1	$1+2=3$	$3+2=5$
p(앞면)	균일	0에 가깝다	0.2에 가깝다

TurboQuant

- **TurboQuant**
  - KV Cache를 더 적은 메모리로 저장
  - 연산 과정에서의 메모리 문제를 해결



TurboQuant mse

- **TurboQuant mse**
  - Objective: minimize MSE

Algorithm 1 TurboQuant$_{\texttt{mse}}$: optimized for MSE

input: dimension d and bit-width b

// Global Parameters for Setting up TurboQuant$_{\texttt{mse}}$

Generate a random rotation matrix $\boldsymbol{\Pi} \in \mathbb{R}^{d \times d}$

Construct codebook by finding centroids $c_1, c_2, \ldots c_{2^b} \in [-1, 1]$ that minimize MSE cost in Eq. (4)

Procedure Quant$_{\texttt{mse}}(\boldsymbol{x})$

$\boldsymbol{y} \leftarrow \boldsymbol{\Pi} \cdot \boldsymbol{x}$

$\texttt{idx}_j \leftarrow \arg\min_{k \in [2^b]} |\boldsymbol{y}_j - c_k|$ for every $j \in [d]$ {$\texttt{idx}_j$'s are b-bit integers}

output: idx

Procedure DeQuant$_{\texttt{mse}}(\texttt{idx})$

$\tilde{\boldsymbol{y}}_j \leftarrow c_{\texttt{idx}_j}$ for every $j \in [d]$

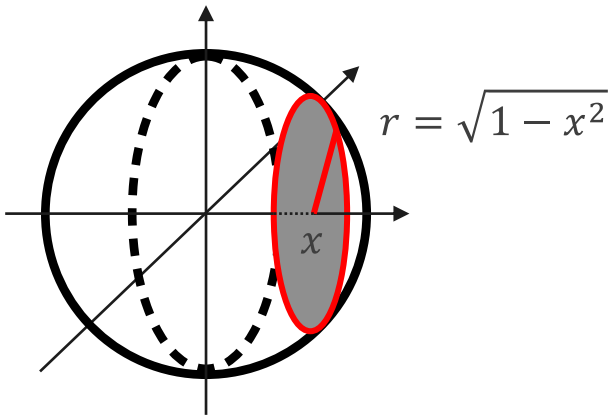
$\tilde{\boldsymbol{x}} \leftarrow \boldsymbol{\Pi}^\top \cdot \tilde{\boldsymbol{y}}$

output: $\tilde{\boldsymbol{x}}$

TurboQuant mse

Global Parameter Setting

- Generate Random Rotate Matrix $\prod \in \mathbb{R}^{d \times d}$



반지름이 $\sqrt{1-x^2}$인 d-1차원 구의 표면적

$$x_j \sim f_X(x) := \frac{\frac{2\pi^{(d-1)/2}}{\Gamma((d-1)/2)}(1-x^2)^{(d-2)/2}}{\frac{2\pi^{d/2}}{\Gamma(d/2)}} \frac{1}{\sqrt{1-x^2}} = \frac{\Gamma(d/2)}{\sqrt{\pi}\Gamma((d-1)/2)}(1-x^2)^{(d-3)/2}$$

좌표 변환에 의한 Jacobian 보정

d-1차원 구의 부피

- Lemma 1. 임의의 양수 d에 대해서 $d-1$ 차원의 단위 구 위의 균일하게 분포된 랜덤 변수 x의 각 좌표는 위와 같은 Beta distribution을 따른다. 또한 d가 커지면 beta distribution는 다음과 같은 정규 분포로 수렴한다.

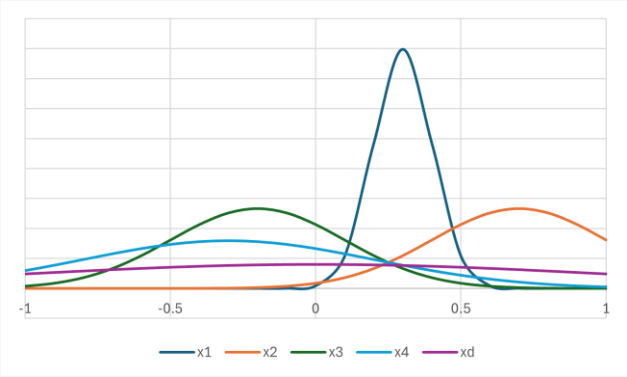
$$f_X(x) \to \mathcal{N}(0, \frac{1}{d})$$

TurboQuant mse

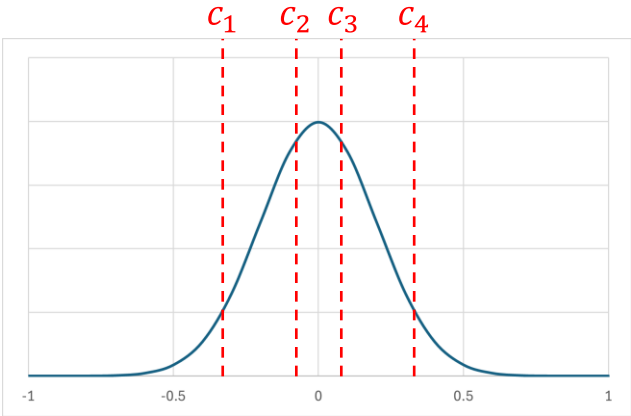
Global Parameter Setting

- Construct codebook $c_1, c_2, \ldots, c_{2^b} \in [-1, 1]$

$$\mathcal{C}(f_X, b) := \min_{-1 \le c_1 \le c_2 \le \cdots \le c_{2^b} \le 1} \sum_{i=1}^{2^b} \int_{\frac{c_{i-1}+c_i}{2}}^{\frac{c_i+c_{i+1}}{2}} |x - c_i|^2 f_X(x) dx$$



Random Rotation



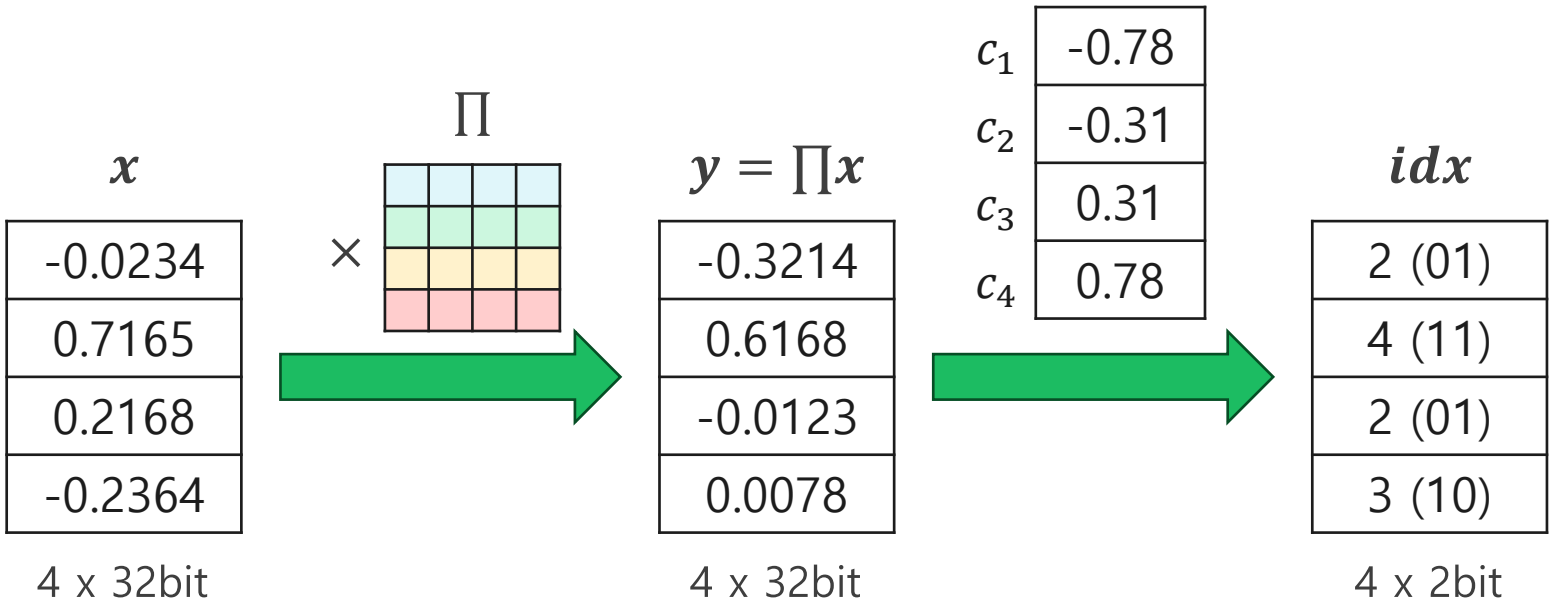
x_1	x_2	x_3	x_4	...	x_d

x'_1	x'_2	x'_3	x'_4	...	x'_d

TurboQuant mse

Quantization

- 동일한 Centroids를 활용하여 회전된 벡터의 각 차원의 Centroid index를 저장

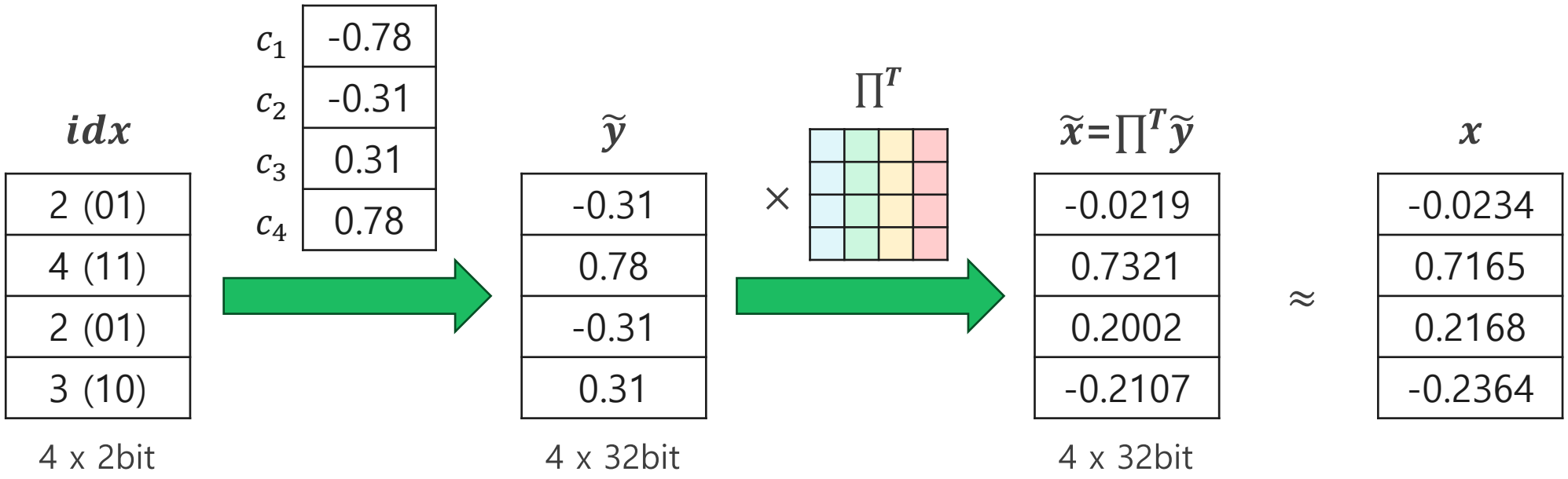


4: **Procedure** $\text{QUANT}_{\texttt{mse}}(\boldsymbol{x})$
5: $\boldsymbol{y} \leftarrow \boldsymbol{\Pi} \cdot \boldsymbol{x}$
6: $\texttt{idx}_j \leftarrow \arg\min_{k \in [2^b]} |\boldsymbol{y}_j - c_k|$ for every $j \in [d]$ {$\texttt{idx}_j$'s are b-bit integers}
7: **output:** idx

TurboQuant mse

Reconstruct

- 동일한 Centroids를 활용하여 회전된 벡터의 각 차원의 Centroid index를 저장



Procedure DEQUANT$_{\texttt{mse}}$(idx)
$\tilde{\boldsymbol{y}}_j \leftarrow c_{\texttt{idx}_j}$ for every $j \in [d]$
$\tilde{\boldsymbol{x}} \leftarrow \boldsymbol{\Pi}^\top \cdot \tilde{\boldsymbol{y}}$
output: $\tilde{\boldsymbol{x}}$

TurboQuant mse

Lemma 3. SLB for random point on hyper sphere

- $d-1$ 차원의 단위 구 위의 균일하게 분포된 랜덤 변수 $x \in \mathbb{S}^{d-1}$에 대한 Lemma 2에서의 B bit에 의한 MSE 오차 함수 D(B)는 다음과 같은 하한이 성립한다.

$$\mathrm{D}(B) \geq 2^{-2B/d}$$

- Proof)
  - A_d를 d-1 차원 구의 표면적이라고 하면, 균일한 분포의 entropy $h(x) = \log_2 A_d$ 이다.
  - Lemma 2의 $h(x)$에 이를 대입하면 다음을 얻을 수 있다.

$$D(p_X, B) \geq \frac{d}{2\pi e} A_d^{2/d} 2^{(-2B/d)}$$

  - Gamma 함수에 대한 Stirling's approximation 공식을 사용하면 다음과 같다.

$$A_d = \frac{2\pi^{d/2}}{\Gamma(\frac{d}{2})} \geq \left(\frac{2\pi e}{d}\right)^{\frac{d}{2}} \sqrt{\frac{2d}{\pi}} (1 - O(\frac{1}{d}))$$

TurboQuant mse

Lemma 3. SLB for random point on hyper sphere

- Proof)
  - A_d에 $2/d$ 승을 취하면 다음과 같이 나타낼 수 있다.

$$A_d^{2/d} \geq \left(\frac{2\pi e}{d}\right)\left(\sqrt{\frac{2d}{\pi}}\left(1 - O\left(\frac{1}{d}\right)\right)\right)^{2/d}$$

  - 여기서 $2/d$ 승 부분은 d가 커짐에 따라 1로 수렴한다. 따라서 다음과 같이 $A_d^{2/d}$의 범위를 구할 수 있다.

$$A_d^{2/d}\rho \gtrapprox \frac{2\pi e}{d}$$

  - 따라서 D(B)는 다음과 같은 하한선을 갖는다.

$$D(p_X, B) \geq \frac{d}{2\pi e} A_d^{2/d} 2^{(-2B/d)} \gtrapprox 2^{(-2B/d)}$$

TurboQuant mse

Theorem 1. Performance guarantee: TurboQuant mse

- 임의의 bit-width $b \geq 1$와 임의의 벡터 $x \in \mathbb{S}^{d-1}$에 대해서, TurboQuant mse로 압축 및 복원한 벡터의 MSE 복원 오차는 다음과 같은 범위를 갖는다.

$$D_{mse} := \mathbb{E}_{\tilde{x}}[\|x - \tilde{x}\|_2^2] \leq \frac{\sqrt{3}\pi}{2}\frac{1}{4^b}$$

- Proof)
  - 원본 벡터 x와 복원 벡터 $\tilde{x}$ 사이의 MSE error는 회전된 벡터 y와 $\tilde{y}$사이의 거리와 동일하기 때문에 다음과 같이 계산할 수 있음.

$$\begin{aligned} D_{mse} &:= \mathbb{E}_{\tilde{x}}[\|x - \tilde{x}\|_2^2] \\ &= \mathbb{E}[\|y - \tilde{y}\|_2^2] = \sum_{j \in [d]} \mathbb{E}[\left|y_j - c_{idx_j}\right|^2] \end{aligned}$$

  - 회전 변환에 의해 모든 좌표가 동일한 분포를 가지므로 모든 차원에 대한 평균은 특정 차원의 d배와 동일한 값을 가짐

$$D_{mse} = d\mathbb{E}[\left|y_1 - c_{idx_1}\right|^2]$$

TurboQuant mse

Theorem 1. Performance guarantee: TurboQuant mse

- Proof)

  - 좌표 하나의 최적의 MSE는 centroid들을 결정하는 식인 $\mathcal{C}(f_X, b)$와 같음. 따라서 MSE 오차 D_{mse}는 다음과 같음

$$D_{mse} = d \cdot \mathcal{C}(f_X, b)$$

  - 고차원에서 f_X의 분포가 정규분포에 수렴하기 때문에, 정규분포 $\mathcal{N}(0, \frac{1}{d})$에 대한 $\mathcal{C}(f_X, b)$ 값은 다음과 같다.
    (Panter-Dite high-resolution formula)

$$\mathcal{C}(f_X, b) \leq \frac{1}{12}\left(\int f_X(x)^{1/3} dx\right)^3 \cdot \frac{1}{4^b} = \frac{\sqrt{3}\pi}{2d}\frac{1}{4^b}$$

  - 따라서, MSE 오차 D_{mse} 는 다음과 같다.

$$D_{mse} \leq \frac{\sqrt{3}\pi}{2}\frac{1}{4^b}$$

TurboQuant prod

TurboQuant prod

- Objective: minimize Inner product error

Algorithm 2 TurboQuant$_{\texttt{prod}}$: optimized for inner product

input: dimension d and bit-width b

// Global Parameters for Setting up TurboQuant$_{\texttt{prod}}$

Instantiate a TurboQuant$_{\texttt{mse}}$ with bit-width $b-1$ as per Algorithm 1

Generate a random projection matrix $\boldsymbol{S} \in \mathbb{R}^{d\times d}$ with i.i.d. entries $\boldsymbol{S}_{i,j} \sim \mathcal{N}(0,1)$

Procedure Quant$_{\texttt{prod}}(\boldsymbol{x})$

$\texttt{idx} \leftarrow \text{Quant}_{\texttt{mse}}(\boldsymbol{x})$

$\boldsymbol{r} \leftarrow \boldsymbol{x} - \text{DeQuant}_{\texttt{mse}}(\texttt{idx})$ {residual vector}

$\texttt{qjl} \leftarrow \texttt{sign}(\boldsymbol{S} \cdot \boldsymbol{r})$ {QJL on residual vector}

output: $(\texttt{idx}, \texttt{qjl}, \|\boldsymbol{r}\|_2)$

Procedure DeQuant$_{\texttt{prod}}(\texttt{idx}, \texttt{qjl}, \gamma)$

$\tilde{\boldsymbol{x}}_{\texttt{mse}} \leftarrow \text{DeQuant}_{\texttt{mse}}(\texttt{idx})$

$\tilde{\boldsymbol{x}}_{\texttt{qjl}} \leftarrow \frac{\sqrt{\pi/2}}{d} \cdot \gamma \cdot \boldsymbol{S}^\top \cdot \texttt{qjl}$

output: $\tilde{\boldsymbol{x}}_{\texttt{mse}} + \tilde{\boldsymbol{x}}_{\texttt{qjl}}$

TurboQuant prod

- **QJL: 1-bit inner product quantization (Quantized Johnson-Lindenstrauss)**
  - 임의의 양의 정수 d에 대해서 QJL은 다음과 같이 정의된다. 이때 $S \in \mathbb{R}^{d \times d}$는 정규분포 $\mathcal{N}(0,1)$ 에서 i.i.d으로 샘플링된 random 행렬이다.

$$Q_{qjl}(x) \coloneqq \text{sign}\,(S \cdot x) \quad for\ any\ x \in \mathbb{R}^d$$

$$Q_{qjl}^{-1}(z) \coloneqq \frac{\sqrt{\pi/2}}{d} S^T z \quad for\ any\ z \in \{-1, +1\}^d$$

- **Lemma 4. Performance guarantee: QJL**
  - 임의의 벡터 $x \in \mathbb{S}^{d-1}$와 $y \in \mathbb{R}^d$에 대하여 다음이 성립한다.

$$\mathbb{E}\left[\left\langle y, Q_{qjl}^{-1}\left(Q_{qjl}(x)\right)\right\rangle\right] = \langle y, x\rangle \quad (Unbiased)$$

$$Var\left(\left\langle y, Q_{qjl}^{-1}\left(Q_{qjl}(x)\right)\right\rangle\right) \leq \frac{\pi}{2d} \cdot \|y\|_2^2 \quad (Variance\ Bound)$$

TurboQuant prod

MSE optimization ≠ Inner product error optimization

- MSE Optimization은 Inner product가 더 작게 나오는 bias가 존재한다.
  - Inner product error

$$\langle y, x \rangle - \langle y, \tilde{x} \rangle = \langle y, x - \tilde{x} \rangle$$

  - 만약 $r = x - \tilde{x}$가 낮더라도, y와 r의 방향에 따라서, inner product에 편향이 존재할 수 있다.

$$inner\ product\ error\ bias = \mathbb{E}[\langle y, r \rangle]$$

TurboQuant prod

MSE optimization ≠ Inner product error optimization

예시) 1-bit MSE-optimal TurboQuant

- 1-bit MSE-optimal TurboQuant의 코드북과 Quantization 함수는 다음과 같다.

$$\text{codebook} = \left\{ -\sqrt{\frac{2}{\pi d}}, +\sqrt{\frac{2}{\pi d}} \right\}$$

$$Q_{mse}(x) = \text{sign}(\Pi x), \qquad Q_{mse}^{-1}(z) = \sqrt{\frac{2}{\pi d}} \Pi^T z$$

$$Q_{qjl}^{-1}(z) := \frac{\sqrt{\pi/2}}{d} S^T z$$

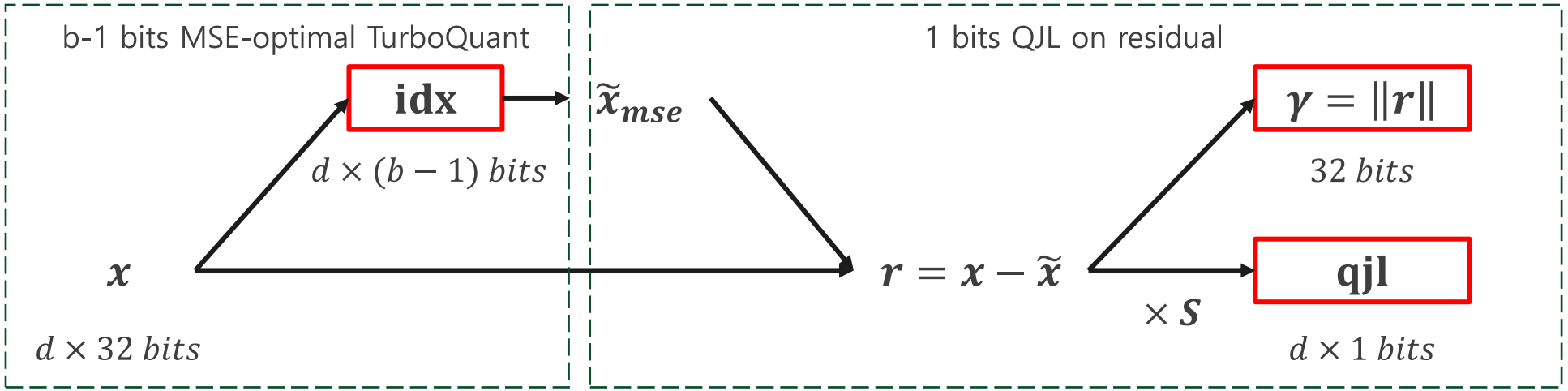
- 따라서, MSE-optimal TurboQuant는 inner product error에 다음과 같은 bias가 존재한다.

$$\mathbb{E}\left[\langle y, Q_{mse}^{-1}(Q_{mse}(x))\rangle\right] = \frac{2}{\pi}\langle y, x\rangle$$

TurboQuant prod

Quantization

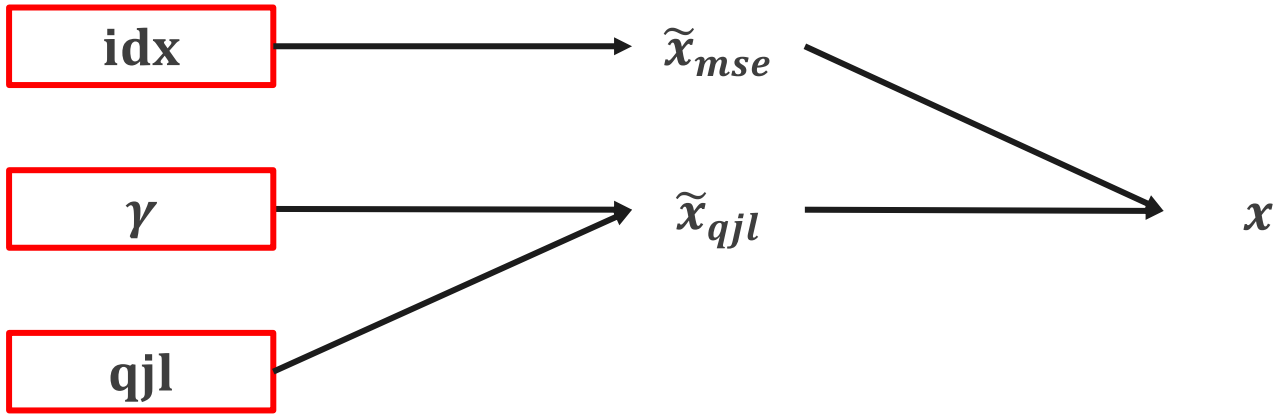
- MSE-optimal TurboQuant로 b-1 bit 압축 (idx)
- 압축 오차 r을 1-bit QJL로 압축 (qjl, γ)



4: **Procedure** $\text{QUANT}_{\text{prod}}(x)$
5: idx $\leftarrow \text{QUANT}_{\text{mse}}(x)$
6: $r \leftarrow x - \text{DEQUANT}_{\text{mse}}(\text{idx})$ {residual vector}
7: qjl $\leftarrow \text{sign}(S \cdot r)$ {QJL on residual vector}
8: **output:** $(\text{idx}, \text{qjl}, \|r\|_2)$

TurboQuant prod

- **Reconstruct**
  - idx 정보로 MSE-optimal TurboQuant 복원 ($\tilde{x}_{mse}$)
  - qjl 벡터와 scale γ로 residual 복원 ($\tilde{x}_{qjl}$)
  - 두 벡터의 합으로 최종 벡터 복원 ($\tilde{x} = \tilde{x}_{mse} + \tilde{x}_{qjl}$)



9: **Procedure** DEQUANT$_{\text{prod}}$(idx, qjl, γ)
10: $\tilde{x}_{\text{mse}} \leftarrow$ DEQUANT$_{\text{mse}}$(idx)
11: $\tilde{x}_{\text{qjl}} \leftarrow \frac{\sqrt{\pi/2}}{d} \cdot \gamma \cdot S^{\top} \cdot \text{qjl}$
12: **output:** $\tilde{x}_{\text{mse}} + \tilde{x}_{\text{qjl}}$

TurboQuant prod

Theorem 2. Performance guarantee: Turboquant prod

- $b \geq 1$ 인 bit-width로 임의의 벡터 $x \in \mathbb{S}^{d-1}$가 존재할 때, Turboquant prod를 통한 압축 결과인 $idx \in [2^{b-1}]^d$와 $qjl \in \{-1,1\}^d$, 그리고 양수 γ를 통해 다시 복원된 벡터 $\tilde{x} \in \mathbb{R}^d$는 다음을 만족한다.

$$\mathbb{E}_{\tilde{x}}[\langle y, \tilde{x} \rangle] = \langle y, x \rangle$$

$$D_{prod} := \mathbb{E}_{\tilde{x}}[|\langle y, x \rangle - \langle y, \tilde{x} \rangle|^2] \leq \frac{\sqrt{3}\pi^2 \cdot \|y\|_2^2}{d} \cdot \frac{1}{4^b}$$

- Proof)
  - $\tilde{x}_{mse}$에 대한 $\langle y, \tilde{x} \rangle$의 조건부 기댓값은 다음과 같다.

$$\begin{aligned}\mathbb{E}[\langle y, \tilde{x} \rangle | \tilde{x}_{mse}] &= \mathbb{E}_{\tilde{x}_{qjl}}\left[\langle y, \tilde{x}_{mse} + \tilde{x}_{qjl} \rangle \middle| \tilde{x}_{mse}\right] \\ &= \langle y, \tilde{x}_{mse} \rangle + \mathbb{E}_{\tilde{x}_{qjl}}\left[\langle y, \tilde{x}_{qjl} \rangle \middle| \tilde{x}_{mse}\right] \\ &= \langle y, \tilde{x}_{mse} \rangle + \langle y, r \rangle = \langle y, x \rangle\end{aligned}$$

  - 최종적으로 기댓값을 계산하면 다음과 같다.

$$\mathbb{E}_{\tilde{x}}[\langle y, \tilde{x} \rangle] = \mathbb{E}_{\tilde{x}_{mse}}\left[\mathbb{E}[\langle y, \tilde{x} \rangle | \tilde{x}_{mse}]\right] = \mathbb{E}[\langle y, x \rangle] = \langle y, x \rangle$$

TurboQuant prod

Theorem 2. Performance guarantee: Turboquant prod

- Proof)
  - $\tilde{x}_{mse}$에 대한 $|\langle y,x\rangle - \langle y,\tilde{x}\rangle|^2$ 의 조건부 기댓값은 다음과 같다.

$$\begin{aligned}\mathbb{E}[|\langle y,x\rangle - \langle y,\tilde{x}\rangle|^2|\tilde{x}_{mse}] &= \mathbb{E}_{\tilde{x}_{qjl}}\left[\left|\langle y,x\rangle - \langle y,\tilde{x}_{\text{mse}}+\tilde{x}_{qjl}\rangle\right|^2\middle|\tilde{x}_{mse}\right] \\ &= \mathbb{E}_{\tilde{x}_{qjl}}\left[\left|\langle y,r\rangle - \langle y,\tilde{x}_{qjl}\rangle\right|^2\middle|\tilde{x}_{mse}\right] \\ &= \mathbb{E}_{\tilde{x}_{qjl}}\left[\left|\mathbb{E}[\langle y,\tilde{x}_{qjl}\rangle] - \langle y,\tilde{x}_{qjl}\rangle\right|^2\middle|\tilde{x}_{mse}\right] \\ &= Var\left(\langle y,\tilde{x}_{qjl}\rangle\middle|\tilde{x}_{mse}\right) \le \frac{\pi}{2d}\cdot\|r\|_2^2\|y\|_2^2\end{aligned}$$

  - 최종적으로 기댓값을 계산하면 다음과 같다.

$$\begin{aligned}D_{prod} &= \mathbb{E}_{\tilde{x}_{mse}}\left[\mathbb{E}[|\langle y,x\rangle - \langle y,\tilde{x}\rangle|^2|\tilde{x}_{mse}]\right] \\ &\le \frac{\pi}{2d}\cdot\|y\|_2^2\mathbb{E}\|x-\tilde{x}_{mse}\|_2^2 \\ &= \frac{\pi}{2d}\cdot\|y\|_2^2 D_{mse} = \frac{\sqrt{3}\pi^2\cdot\|y\|_2^2}{d}\cdot\frac{1}{4^b}\end{aligned}$$

TurboQuant prod

- **Theorem 3. Lower bound on best achievable compression distortion**
  - 임의의 quantization 알고리즘 $Q: \mathbb{S}^{d-1} \to \{0,1\}^{b \cdot d}$와 복원 과정 $Q^{-1}: \{0,1\}^{b \cdot d} \to \mathbb{R}^d$에 대해서 다음을 만족시키는 $x \in \mathbb{S}^{d-1}$, $y \in \mathbb{S}^{d-1}$이 존재한다.

$$D_{mse}(Q) := \mathbb{E}[\|x - Q^{-1}Q(x))\|_2^2] \geq \frac{1}{4^b}$$

$$D_{prod}(Q) = \mathbb{E}[|\langle y, x\rangle - \langle y, Q^{-1}Q(x))\rangle|^2] \geq \frac{1}{d} \cdot \frac{1}{4^b}$$

- **TurboQuant는 Near-Optimal Quantization이다.**
  - distortion이 최소인 방법은 아니지만, 거의 최소인 방법이다. (Near-Optimal)

$$D_{mse} \leq \frac{\sqrt{3}\pi}{2} \frac{1}{4^b}$$

$$D_{prod} = \frac{\sqrt{3}\pi^2 \cdot \|y\|_2^2}{d} \cdot \frac{1}{4^b}$$

Experiments Setting

- **Dataset**
  - **DBpedia Entities dataset**
  - **Needle-In-A-Haystack Test**
  - **LongBench-V1**
- **Model**
  - **OpenAI3 embeddings (1536-dim)**
  - **Llama-3.1-8B-Instruct, Ministral-7B-Instruct**

Error distribution

- **Error distribution (OpenAI3, Dbpedia)**
  - **mse는 bias가 있는 반면에, prod는 없음**

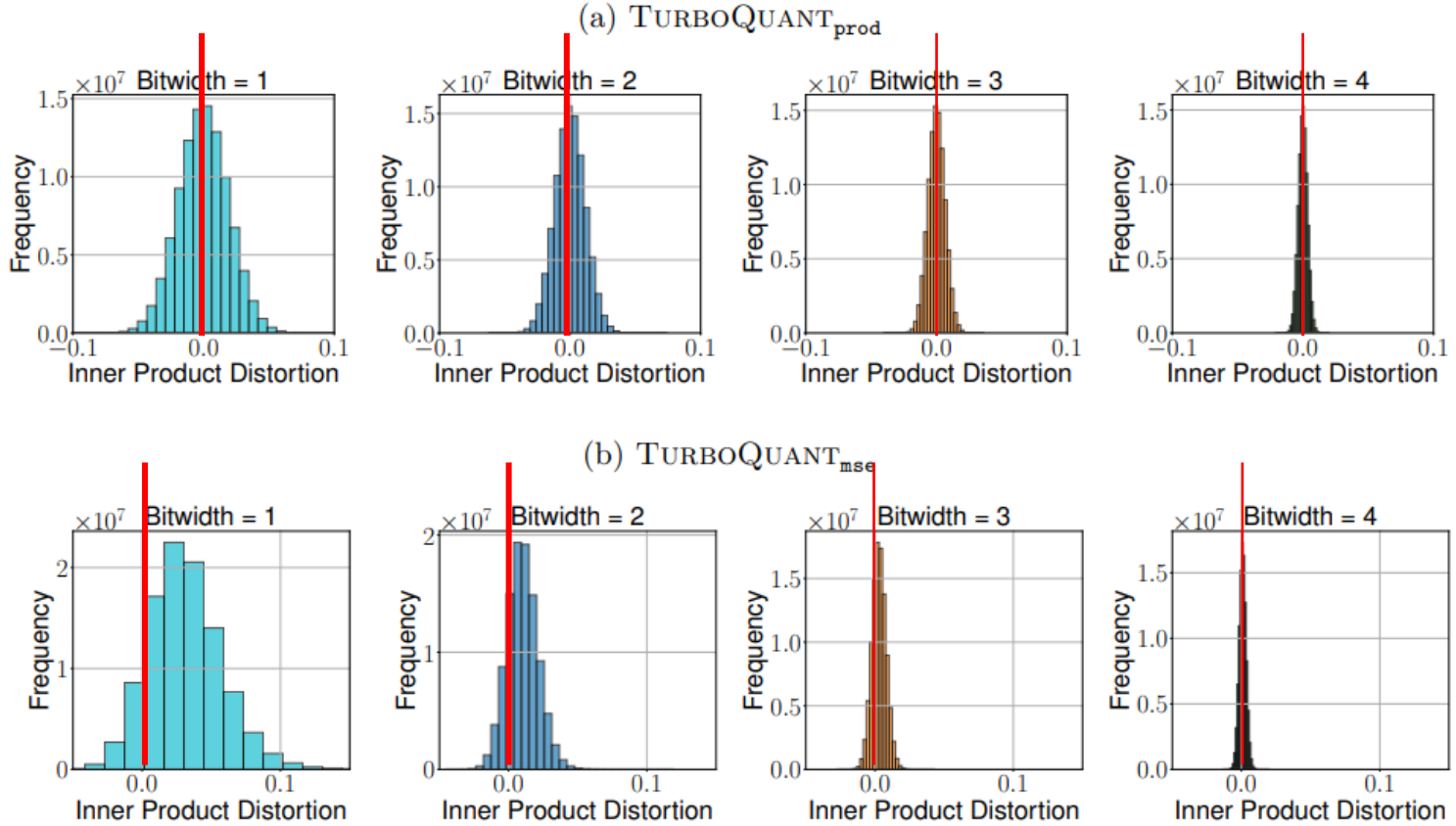
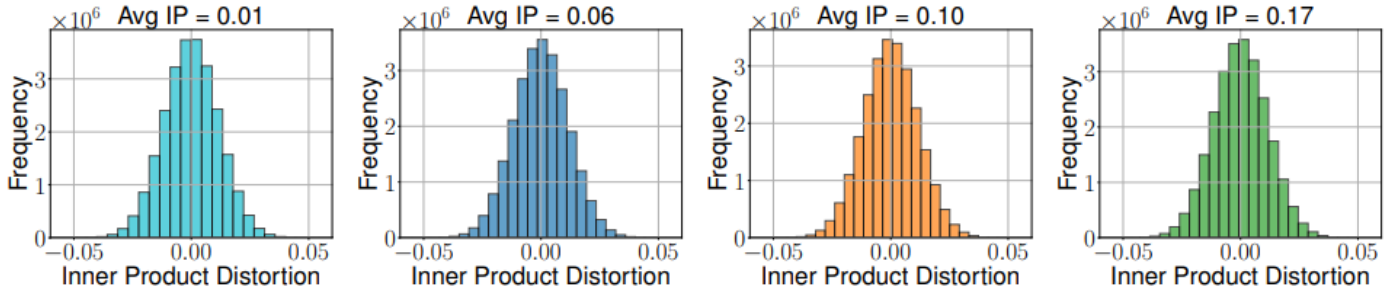


Figure 1: Error distribution of TurboQuant$_{prod}$ and TurboQuant$_{mse}$ for Inner Product Estimation.

Variance of Distortion

- **Variance of Distortion (OpenAI3, Dbpedia)**
  - **Turboquant prod는 분산이 inner product와 무관함**
  - **Turboquant mse는 inner product에 비례하는 bias를 가짐**

(a) TurboQuant$_{\texttt{prod}}$



(b) TurboQuant$_{\texttt{mse}}$

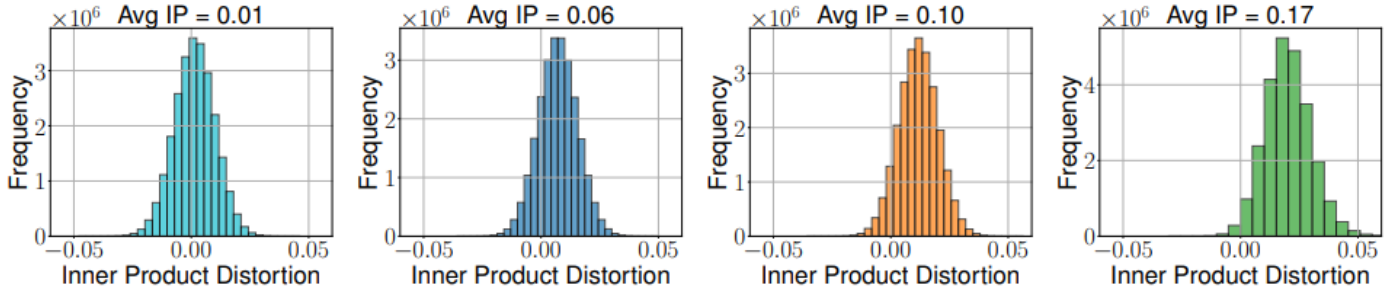


Figure 2: The variance of Inner-product error remains constant for TurboQuant$_{\texttt{prod}}$, while in TurboQuant$_{\texttt{mse}}$ increases with the average inner product. Bit-width is $b = 2$.

Theoretical Bounds

- **Theoretical Inner-product error and MSE**
  - 충분한 bandwidth가 주어진 경우 mse가 더 좋음
  - Bandwidth가 커질수록 이론적 상한선에 수렴

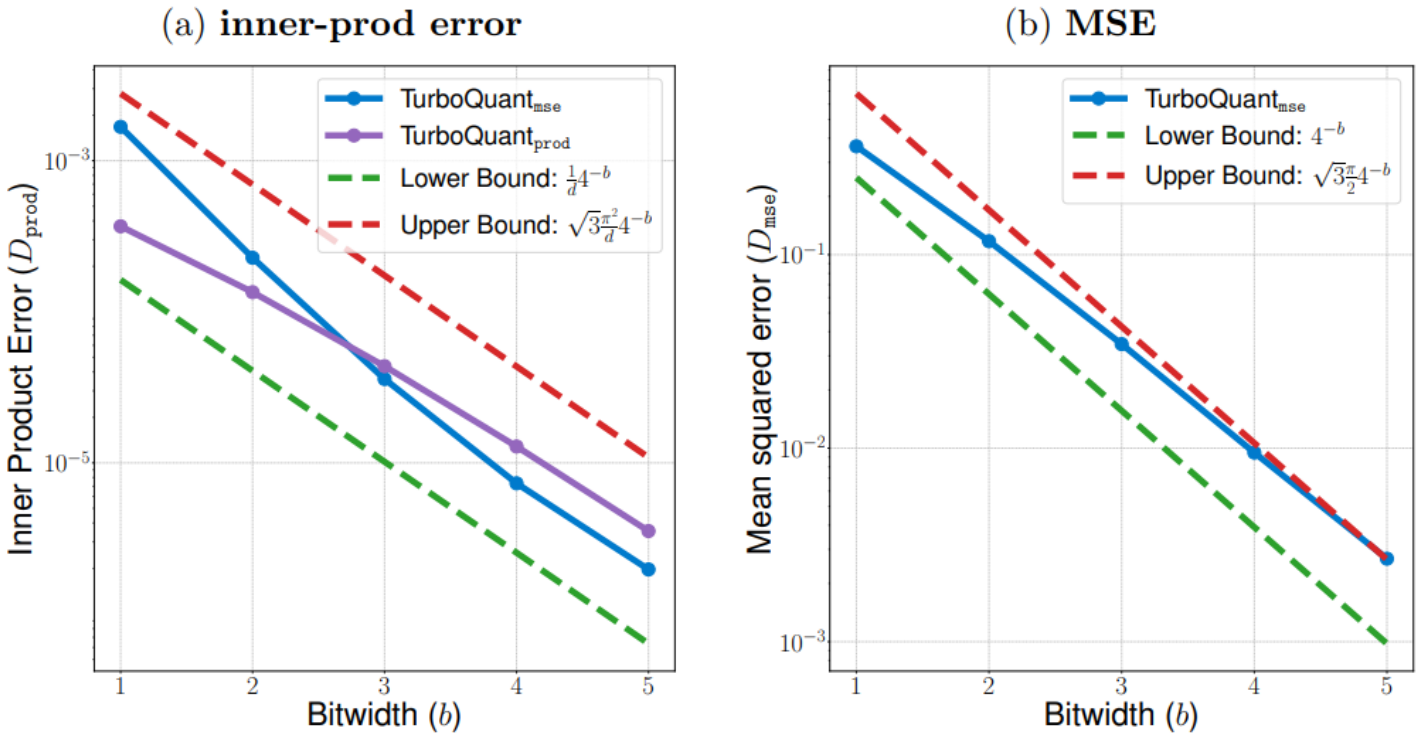


Figure 3: Comparison of inner-product error and MSE against theoretical bounds across different bit ratios.

Needle-In-A-Haystack test

- 긴 token 속에서 답을 찾는 task
  - 4-bit 설정에서 압축하지 않은 모델과 동일한 성능 달성

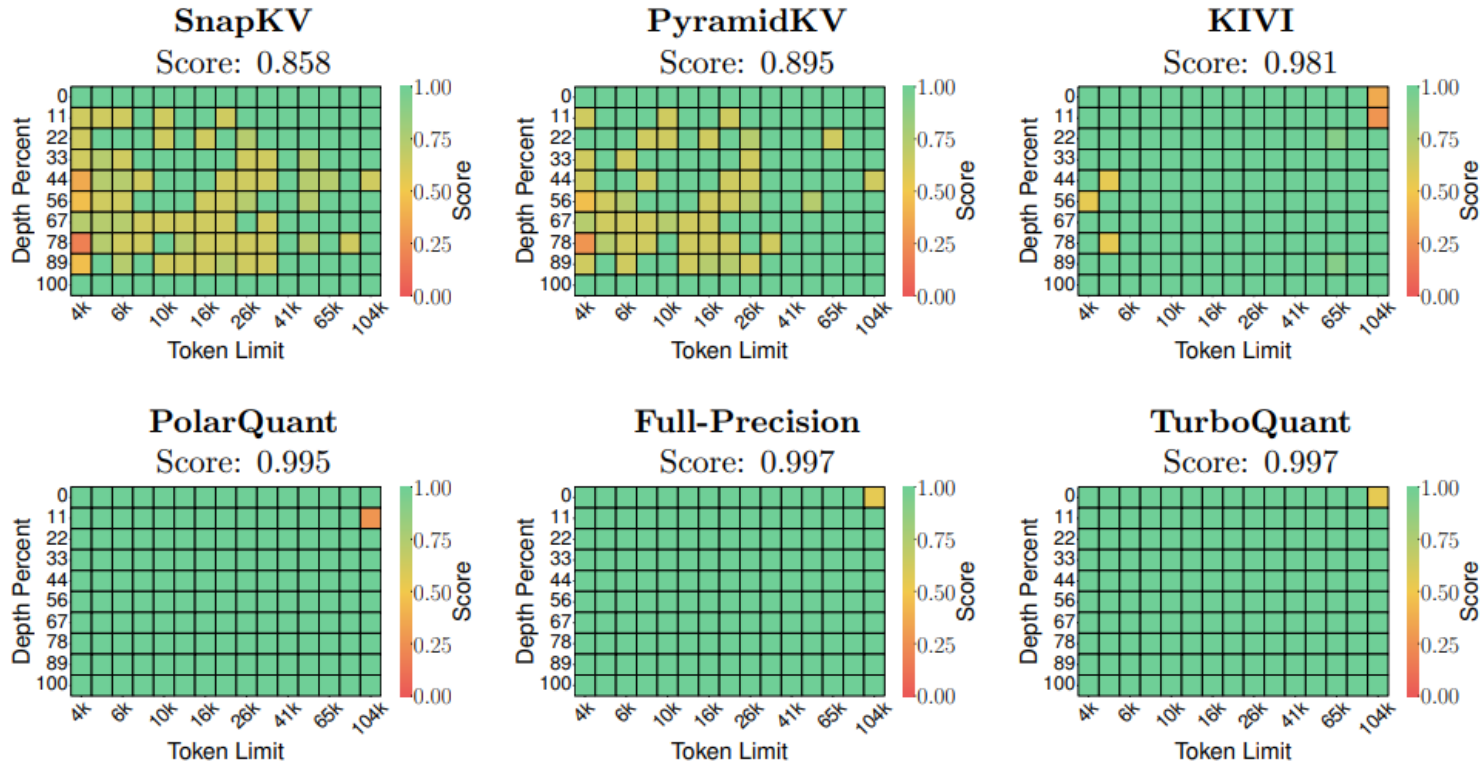


Figure 4: Evaluation of `Llama-3.1-8B-Instruct` on the "Needle-In-A-Haystack" test, where a model must retrieve a hidden sentence from long-context sequences. While some methods struggle with recall, TURBOQUANT, despite being more than 4× quantized, achieves the same exact performance as the uncompressed baseline.

LongBench-V1

- KV Cache를 압축하면 LLM의 성능이 얼마나 유지되는가
  - 6배 적은 KV Cache 메모리로 거의 유사한 성능 달성

Method	KV Size	SingleQA	MultiQA	Summarization	Few shot	Synthetic	Code	Average
				Llama-3.1-8B-Instruct				
Full Cache	16	45.29	45.16	26.55	68.38	59.54	46.28	50.06
KIVI	3	43.38	37.99	27.16	68.38	59.50	44.68	48.50
KIVI	5	45.04	45.70	26.47	68.57	59.55	46.41	50.16
PolarQuant	3.9	45.18	44.48	26.23	68.25	60.07	45.24	49.78
TurboQuant (ours)	2.5	44.16	44.96	24.80	68.01	59.65	45.76	49.44
TurboQuant (ours)	3.5	45.01	45.31	26.00	68.63	59.95	46.17	50.06
				Ministral-7B-Instruct				
Full Cache	16	47.53	49.06	26.09	66.83	53.50	47.90	49.89
TurboQuant (ours)	2.5	48.38	49.22	24.91	66.69	53.17	46.83	49.62

Table 1: LongBench-V1 [10] results of various KV cache compression methods on `Llama-3.1-8B-Instruct`.

Quantization Time & Near Neighbour Search

Quantization Time & Search

Approach	d=200	d=1536	d=3072
Product Quantization	37.04	239.75	494.42
RabitQ	597.25	2267.59	3957.19
TurboQuant	0.0007	0.0013	0.0021

Table 2: Quantization time (in seconds) for different approaches across various dimensions using 4-bit quantization.

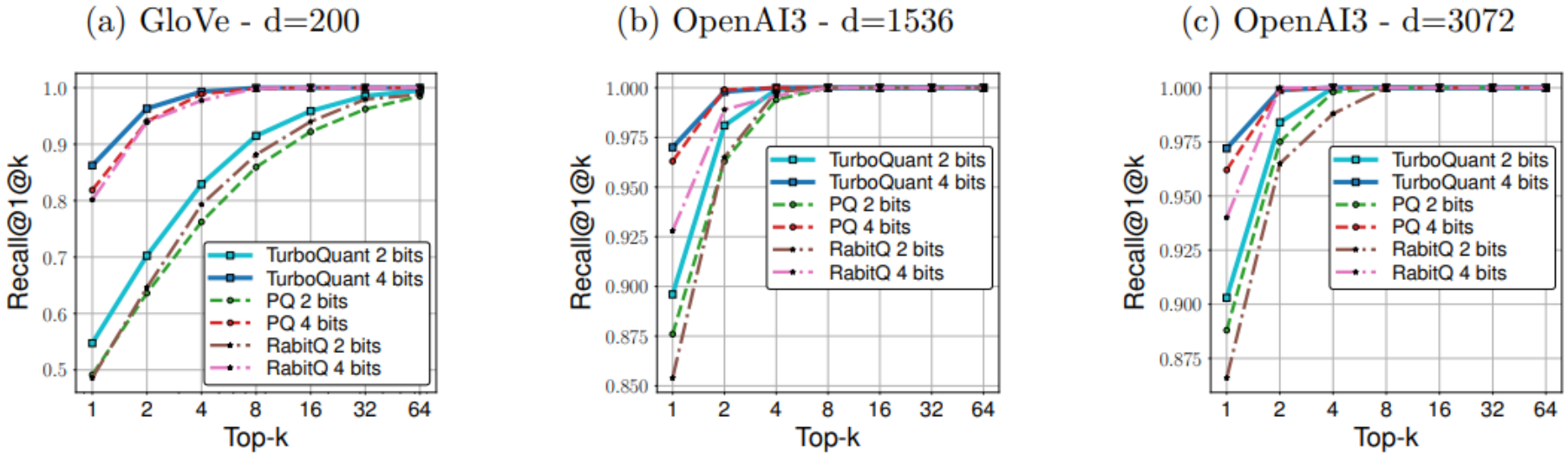


Figure 5: Recall comparison on different datasets with different embedding dimensions.

Conclusion

- **TurboQuant 알고리즘**
  - 고차원 벡터를 효율적으로 압축하기 위한 방법 제시
  - MSE optimization의 bias, QJL의 variance에 대한 지적
  - b-1 bit로 MSE로 압축 후 QJL로 bias를 제거하는 방법을 제안

- **효율적인 압축 방법**
  - 압축을 위한 데이터 학습 X (random matrix 활용)
  - 빠르고 효율적인 압축 과정

Thank you

Dooyoung Kim

(kdysunleo98@gmail.com)

질문 https://forms.gle/LtvyMJ7BFwMKpWtz8 **피드백** https://forms.gle/PAmxLQnRBZVhAMaw8

응답 https://docs.google.com/spreadsheets/d/1uWyc0pUfQOwTTZUDyY3gxlmU5xcFro90kKJKOcDitnk/edit?usp=sharing

Appendix A. Lemma 2 (증명)

- **Lemma 2. Shannon Lower Bound (SLB, Shannon's lossy source coding theorem)**
  - 확률 분포 p_X를 따르고 유한한 미분가능 엔트로피 $h(x)$를 갖는 랜덤 벡터 $x \in \mathbb{R}^d$ 에 대해서, B bit로 압축 후 복원한 벡터 y와의 MSE 오차 $D(p_X, B)$ 는 다음과 같다.

$$D(p_X, B) \coloneqq \inf\{\mathbb{E}[\|x - y\|_2^2] : I(x; y) \leq B\}$$

  - 이때 $D(p_X, B)$ 는 mutual information $I(x; y)$가 최대 B인 모든 x와 y의 결합분포의 infimum(하한)이며, 다음과 같은 값을 갖는다.

$$D(p_X, B) \geq \frac{d}{2\pi e} 2^{\left(\frac{2}{d}\right)(h(x)-B)}$$

Appendix A. Lemma 2 (증명)

Proof)

- 조건 $I(x;y) \leq B$를 풀어서 쓰면 조건부 엔트로피 $h(x|y)$를 다음과 같이 구할 수 있음

$$I(x;y) = h(x) - h(x|y) \leq B$$
$$h(x|y) \geq h(x) - B$$

- 복원 오차 e를 x와 y의 차이로 정의하면 다음이 성립함.

$$x = y + e$$
$$h(x|y) = h(e|y)$$
$$h(e) \geq h(e|y) = h(x|y) \geq h(x) - B$$

- 주어진 MSE에서 엔트로피를 최대화 하는 분포는 Gaussian 분포임. (분산이 σ^2로 고정된 경우, 증명 Appendix @)

- Gaussian 분포를 따르는 $e \sim \mathcal{N}(0, \sigma^2 I)$의 엔트로피는 다음과 같음.

$$h(e) = \frac{d}{2}\log(2\pi e\sigma^2)$$

Appendix A. Lemma 2 (증명)

Proof)

- MSE는 오차 e에 대한 제곱 평균이므로 다음과 같이 표현할 수 있음

$$\mathrm{D} = \mathbb{E}\|e\|_2^2 = d\sigma^2$$

$$\sigma^2 = \frac{D}{d}$$

- 따라서 다음과 같은 관계가 성립함

$$\frac{d}{2}\log(2\pi e\frac{D}{d}) = h(e) \geq h(x) - B$$

- 양변을 정리 후, 지수를 취하면

$$\log(2\pi e\frac{D}{d}) \geq \frac{2}{d}(h(x) - B)$$

$$2\pi e\frac{D}{d} \geq 2^{\frac{2}{d}(h(x)-B)}$$

$$D \geq \frac{d}{2\pi e}2^{\frac{2}{d}(h(x)-B)}$$

Appendix B. 주어진 분산 σ^2 내 최대 엔트로피를 갖는 분포 증명

- **Proof) 주어진 분산 σ^2 내 최대 엔트로피를 갖는 분포는 Gaussian이다.**
  - 연속확률변수 X에 대한 확률 분포 $p(x)$가 다음과 같은 성질을 갖는다.

$$\mathbb{E}[X] = 0, \qquad V(X) = \sigma^2$$

  - 동일한 평균 및 분산을 갖는 Gaussian 분포를 $q(x)$로 정의하면 다음과 같다.

$$q(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

  - $p(x)$와 $q(x)$의 KL Divergence는 다음과 같다.

$$D_{KL}(p||q) = \int p(x) \log \frac{p(x)}{q(x)} dx \geq 0$$

Appendix B. 주어진 분산 σ^2 내 최대 엔트로피를 갖는 분포 증명

- **Proof) 주어진 분산 σ^2 내 최대 엔트로피를 갖는 분포는 Gaussian이다.**
  - 이를 $p(x)$에 대한 엔트로피로 표현하면 다음과 같다.

$$-\int p(x)\log p(x)\,dx \le -\int p(x)\log q(x)\,dx$$

$$h(p) \le -\int p(x)\log q(x)\,dx$$

  - Gaussian $q(x)$에 log를 취하면,

$$\log q(x) = -\frac{1}{2}\log(2\pi\sigma^2) - \frac{x^2}{2\sigma^2}$$

$$-\int p(x)\log q(x)\,dx = -\mathbb{E}_p[\log q(X)] = \frac{1}{2}\log(2\pi\sigma^2) + \frac{\mathbb{E}_p[X^2]}{2\sigma^2}$$

Appendix B. 주어진 분산 σ^2 내 최대 엔트로피를 갖는 분포 (증명)

- **Proof) 주어진 분산 σ^2 내 최대 엔트로피를 갖는 분포는 Gaussian이다.**
  - $q(x)$는 평균이 0, 분산이 σ^2인 분포이므로,

$$\mathbb{E}_p[X^2] = \sigma^2$$

$$-\int p(x) \log q(x)\, dx = \frac{1}{2}\log(2\pi\sigma^2) + \frac{1}{2} = \frac{1}{2}\log(2\pi e\sigma^2)$$

  - 따라서, $q(x)$의 엔트로피 $h(p)$는 다음과 같은 범위를 갖는다.

$$h(q) \leq \frac{1}{2}\log(2\pi e\sigma^2)$$

  - 즉, $q(x)$의 엔트로피는 $q(x)$가 Gaussian 분포일 때 최댓값 $\frac{1}{2}\log(2\pi e\sigma^2)$ 을 갖는다.
    (KL Divergence의 등호 성립조건 = 분포가 동일)

Appendix C. Lemma 4 (증명)

Proof)

- x를 quantization 후 다시 복원한 벡터는 다음과 같다.

$$Q_{qjl}^{-1}\left(Q_{qjl}(x)\right) = \frac{\sqrt{\pi/2}}{d} S^T sign(Sx)$$

- y와 내적을 취한 결과는 다음과 같다.

$$\begin{aligned}\left\langle y, Q_{qjl}^{-1}\left(Q_{qjl}(x)\right)\right\rangle &= \frac{\sqrt{\pi/2}}{d}\langle y, S^T \text{sign}(Sx)\rangle \\ &= \frac{\sqrt{\pi/2}}{d}\langle Sy, \text{sign}(Sx)\rangle\end{aligned}$$

- Random matrix S의 row 단위로 쓰면 다음과 같이 표현할 수 있다.

$$\left\langle y, Q_{qjl}^{-1}\left(Q_{qjl}(x)\right)\right\rangle = \frac{1}{d}\sum_{i=1}^{d}\sqrt{\pi/2}\left(s_i^T y\right)\text{sign}(s_i^T x)$$

Appendix C. Lemma 4 (증명)

Proof)

- 각 row를 하나의 random 변수 z_i로 정의하면, s_i가 서로 독립이기 때문에 z_i들도 서로 독립이고 동일한 분포를 갖는다.

$$z_i = \sqrt{\pi/2}\left(s_i^T y\right)\text{sign}(s_i^T x)$$

- y를 x방향과 수직인 방향으로 나누면, $y = \langle y, x\rangle x + y_\perp$이고, 이를 활용하면 다음과 같다.

$$\begin{aligned}\mathbb{E}\left[\left(s_i^T y\right) sign\left(s_i^T x\right)\right] &= \langle y, x\rangle \mathbb{E}\left[\left(s_i^T x\right) sign\left(s_i^T x\right)\right] + \mathbb{E}\left[\left(s_i^T y_\perp\right) sign(s_i^T x)\right] \\ &= \langle y, x\rangle \mathbb{E}\left[\left(s_i^T x\right) sign\left(s_i^T x\right)\right] + 0 = \langle y, x\rangle \mathbb{E}\left[\left|s_i^T x\right|\right]\end{aligned}$$

- $s_i \sim \mathcal{N}(0, I)$이기 때문에 다음이 성립한다.

$$\mathbb{E}\left[\left|s_i^T x\right|\right] = \sqrt{2/\pi}$$

- 따라서 z_i들의 평균을 구하면 다음과 같다.

$$\left\langle y, Q_{qjl}^{-1}\left(Q_{qjl}(x)\right)\right\rangle = \frac{1}{d}\sum_{i=1}^{d}\mathbb{E}[z_i] = \langle y, x\rangle \quad (Unbiasedness)$$

Appendix C. Lemma 4 (증명)

- **Proof)**
  - z_i의 분산을 계산을 계산하면 다음과 같다.

$$Var(z_i) = \frac{\pi}{2} Var\left(\left(s_i^T y\right)\text{sign}\left(s_i^T x\right)\right)$$

  - 분산은 항상 제곱 평균보다 작으므로 다음이 성립한다.

$$Var\left(\left(s_i^T y\right)\text{sign}\left(s_i^T x\right)\right) \leq \mathbb{E}[\left(s_i^T y\right)^2 \text{sign}\left(s_i^T x\right)^2]$$

  - $\text{sign}\left(s_i^T x\right)^2$은 항상 1이기 때문에 다음이 성립한다.

$$\mathbb{E}\left[\left(s_i^T y\right)^2 \text{sign}\left(s_i^T x\right)^2\right] = \mathbb{E}\left[\left(s_i^T y\right)^2\right]$$

  - $s_i \sim \mathcal{N}(0, I)$이므로 $s_i^T y \sim \mathcal{N}(0, \|y\|_2^2)$이고 분산의 상한선은 다음과 같다.

$$Var\left(\left\langle y, Q_{qjl}^{-1}\left(Q_{qjl}(x)\right)\right\rangle\right) = \frac{1}{d^2}\sum_{i=1}^{d} Var(z_i) \leq \frac{\pi}{2d} \cdot \|y\|_2^2 \qquad (Variance\ Bound)$$