Reformer

The Efficient Transformer

Published as a conference paper at ICLR 2020
Kitaev, Nikita, Łukasz Kaiser, and Anselm Levskaya. "Reformer: The efficient transformer." ***arXiv preprint arXiv:2001.04451*** **(2020).**

Problems & Solutions

Large-scale long-sequence models yield great results but strain resources to the point where some argue that this trend is breaking NLP research.

- **Attention on sequences of length L is $O(L^2)$ in both computational and memory complexity**

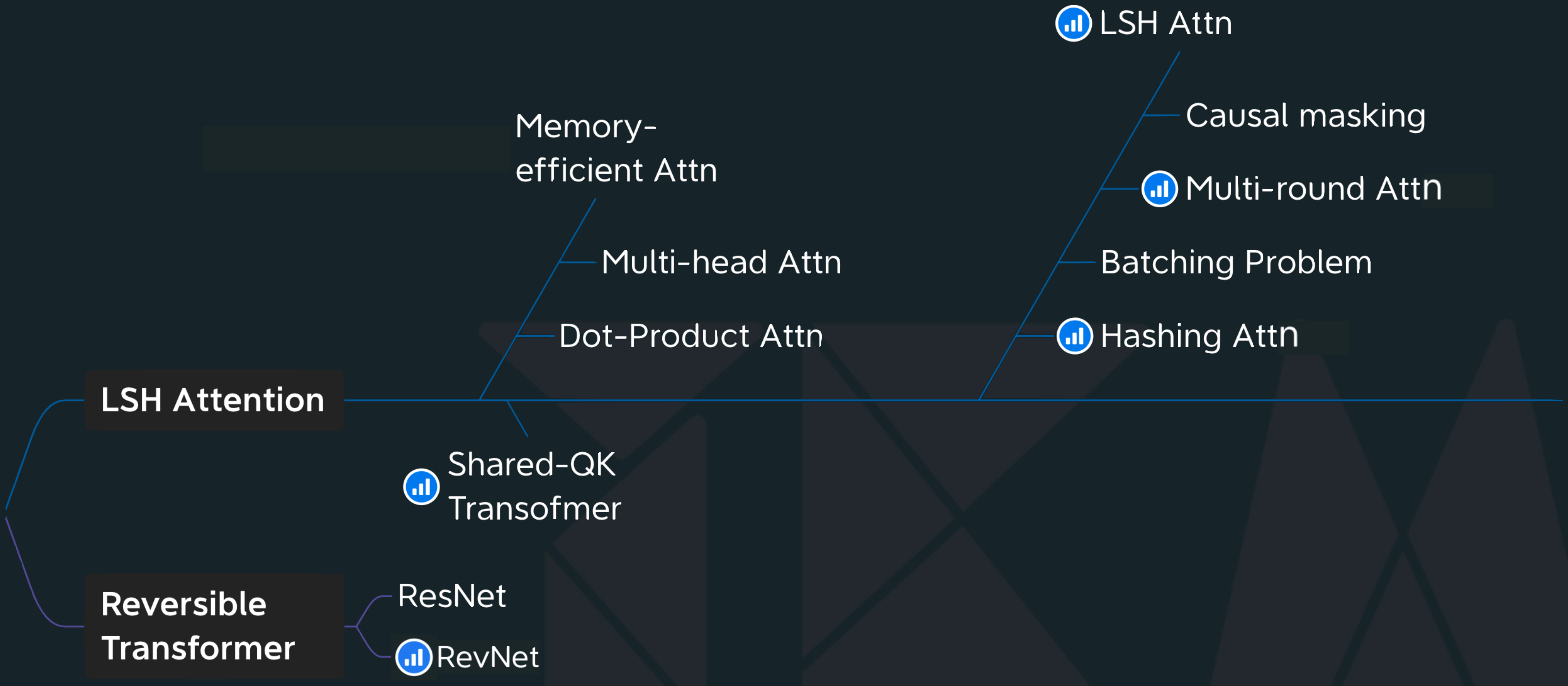
  → LSH Attention.

  i.e. Let Batch size = 1, Seq length S = 64K:
  In the original attention the QK^T term would cost 1 * 64K * 64K = 16G Memory (in float-32).

- **Memory in a model with N layers is N-times larger**

  → Reversible Residual Network.

Road Map


LSH Attention
Memory-efficient Attn
Multi-head Attn
Dot-Product Attn
Shared-QK Transofmer
LSH Attn
Causal masking
Multi-round Attn
Batching Problem
Hashing Attn
Reversible Transformer
ResNet
RevNet

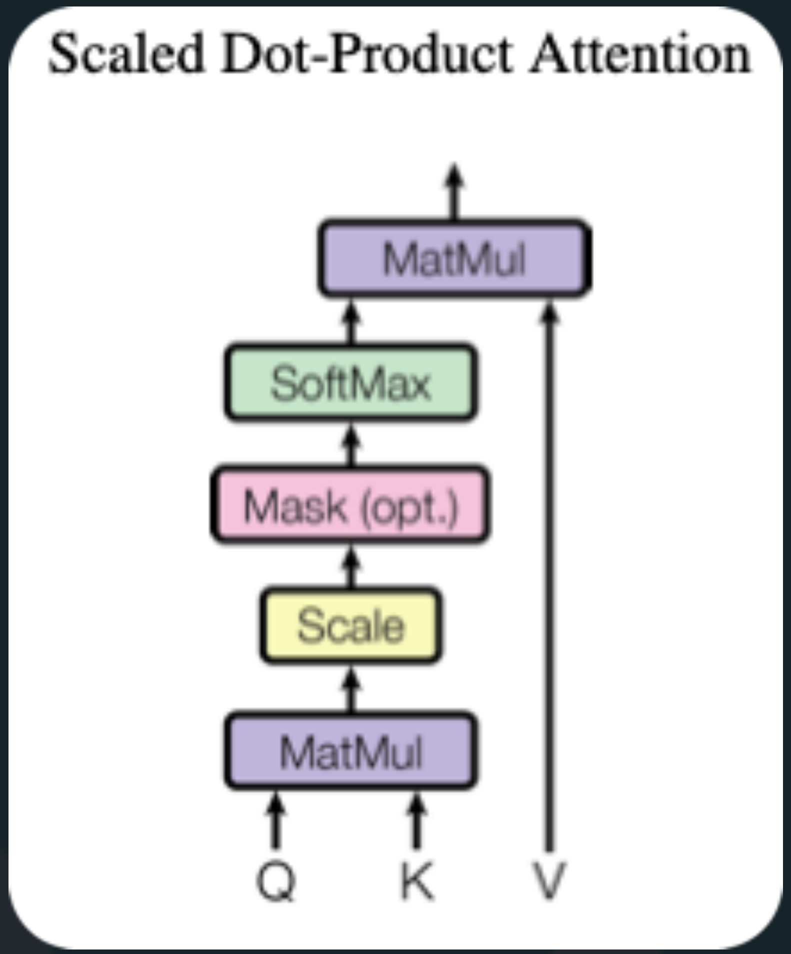
Dot-product attention & Multi-head attention

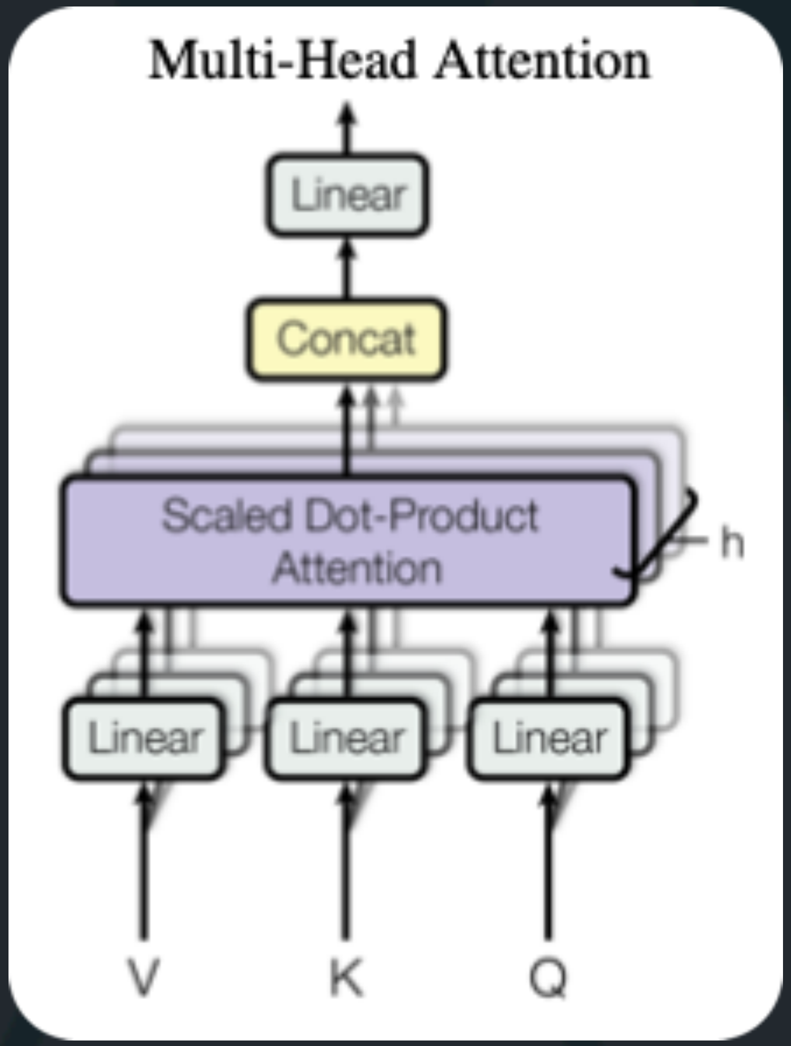
- **Dot-product attention** ($Q : [B, S, d_k]$, $K^T : [B, d_k, S]$, $V : [B, S, d_v]$)

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad : [B, S, S] \rightarrow [B, S, d_v]$$

- **Multi-head attention (h = number of heads)**

$$\begin{aligned} \text{MHAttn}(Q, K, V) &= \text{cat}(\text{head}_1, \ldots, \text{head}_h)W^O \\ \text{head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \end{aligned}$$





[1] Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

Memory-efficient attention & Shared-QK transformer

- **Memory-efficient attention (Separately computing** q_i : $[B,1,d_k]$, K^T : $[B, d_k, S]$, V : $[B, S, d_v]$**)**

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{q_i K^T}{\sqrt{d_k}}\right)V \quad : [B,1,S] \rightarrow [B,1,d_v]$$

- **Shared-QK Transformer**

$$W^Q = W^K$$

Hashing attention & Locality sensitive hashing

- **Hashing attention**

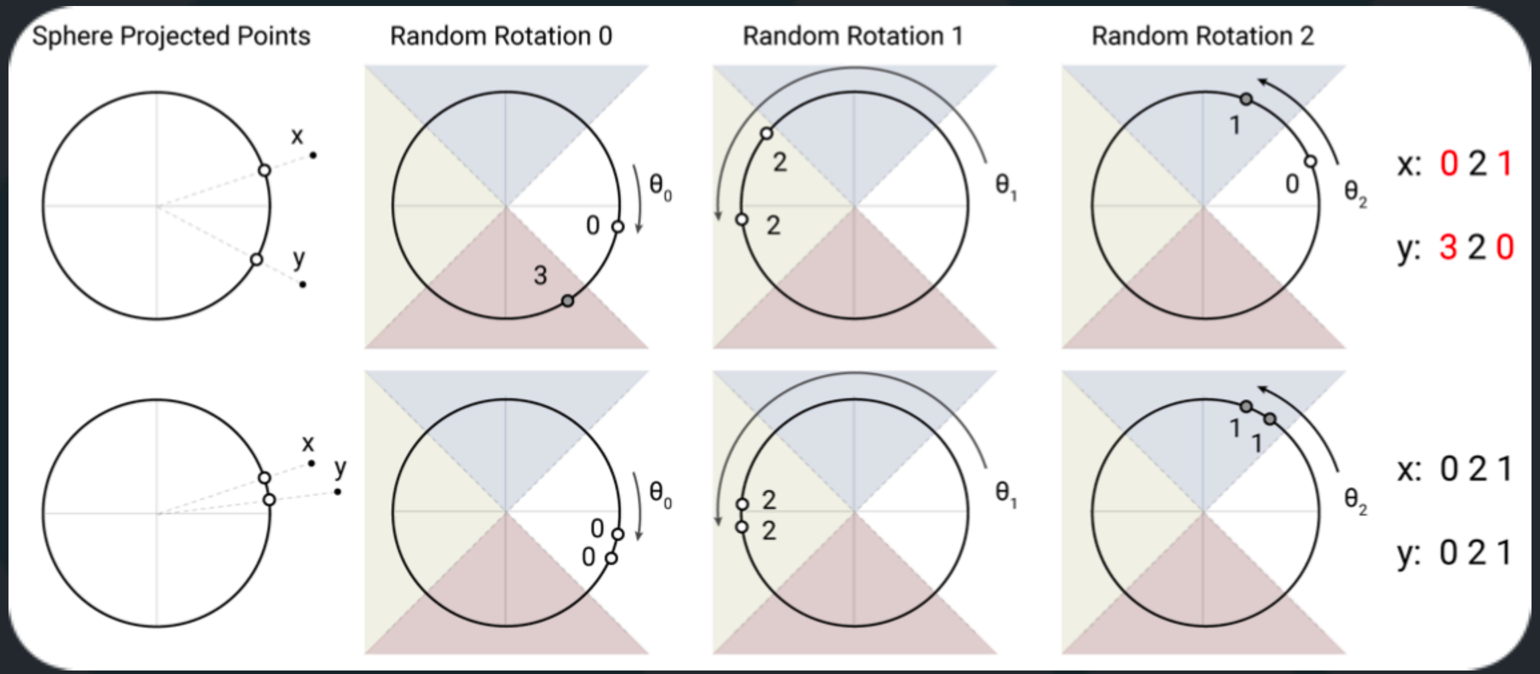
$$\text{softmax}(z) = \frac{e^{z_i}}{\sum_{k=1}^{|z|} e^{z_k}} \ \forall\ i = 1,\dots,|z|$$

Since softmax is dominated by the largest elements,
for each query q_i we only need to focus on the keys in K that are closest to q_i.

- **LSH (Locality-Sensitive Hashing)**

Implying random projections:

fix random matrix R with size $[d_k, b/2]$ to get b hashes.



Hashing function [3]:

$h(x) = \arg\max([xR; -xR])$, where $[u; v]$ denotes the concatenation of two vectors.

[3] Andoni, Alexandr, et al. "Practical and optimal LSH for angular distance." *Advances in neural information processing systems* 28 (2015).

LSH Attention

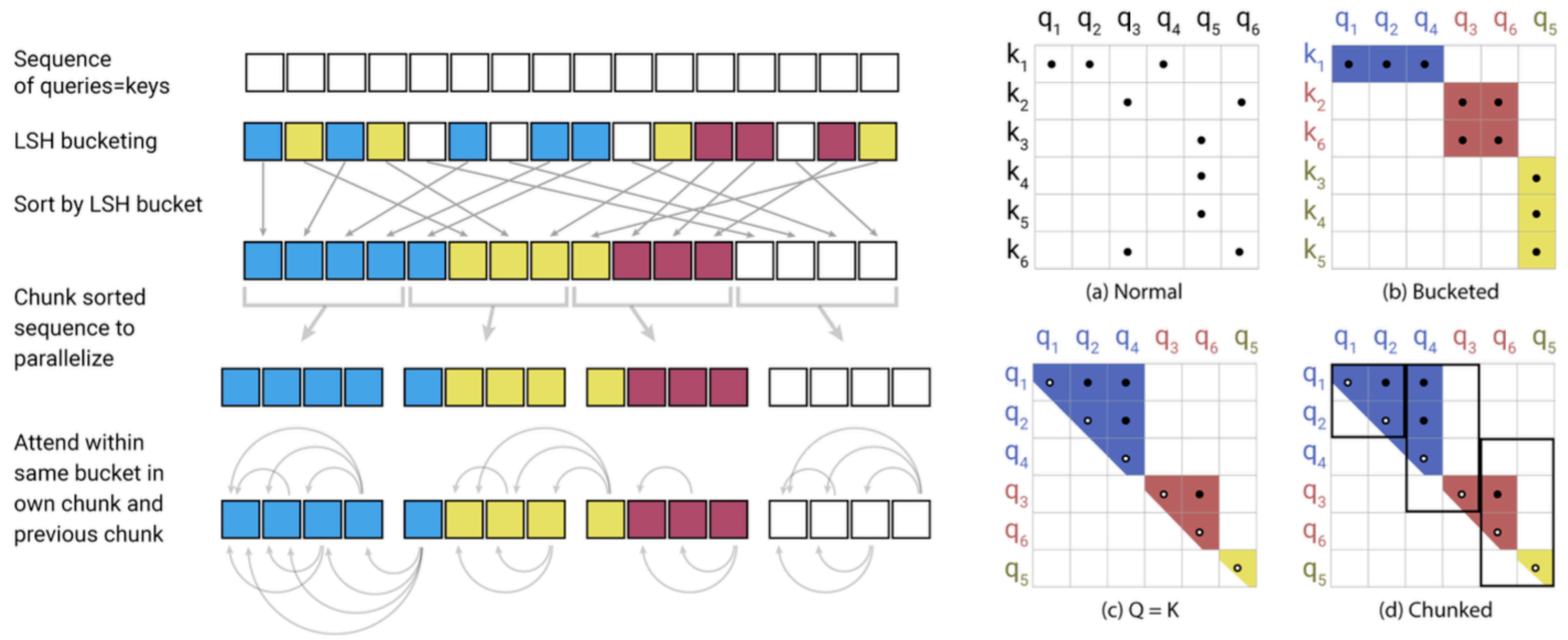


Figure 2: Simplified depiction of LSH Attention showing the hash-bucketing, sorting, and chunking steps and the resulting causal attentions. (a-d) Attention matrices for these varieties of attention.

Formalize the normal attention

- **Rewrite** the normal attention, for a single query position i at a time (omit scaling $\sqrt{d_k}$).

$$o_i = \sum_{j \in \mathcal{P}_i} \frac{\exp(q_i \cdot k_j)}{\sum_{j \in \mathcal{P}_i} \exp(q_i \cdot k_j)} v_j$$

$$= \sum_{j \in \mathcal{P}_i} \exp(q_i \cdot k_j - z(i, \mathcal{P}_i))\ v_j$$

$$\text{where } \begin{cases} \mathcal{P}_i = \{j : j \leq i\} \\ z \ = \text{partition function} \end{cases}$$

- **For batching, perform attention over a larger set $\tilde{\mathcal{P}}_i = \{0,1,\ldots,l\} \supseteq \mathcal{P}_i$, where l is sequence length.**

$$o_i = \sum_{j \in \tilde{\mathcal{P}}_i} \exp(q_i \cdot k_j - m(j, \mathcal{P}_i) - z(i, \mathcal{P}_i))\ v_j$$

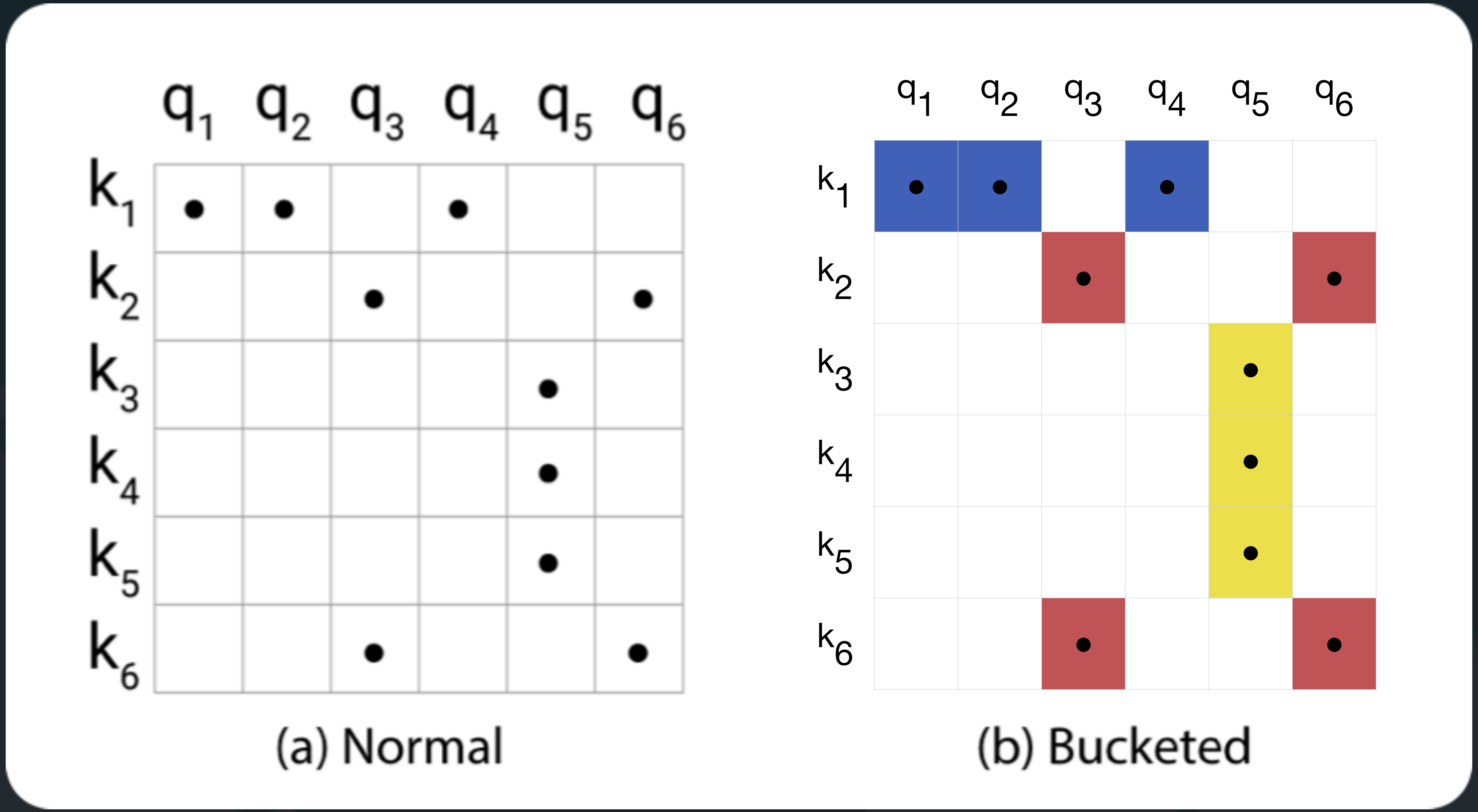
$$\text{where } m(j, \mathcal{P}_i) = \begin{cases} \infty, \text{ if } j \notin \mathcal{P}_i \\ 0\ , \text{ if o.w.} \end{cases}$$

LSH Attention

- **In LSH Attention the $\mathscr{P}_i = \{j : h(q_i) = h(k_j)\}$ (the dots)**

  (a) Normal: Original attention process.

  (b) Bucketed: Sorting by hash number.



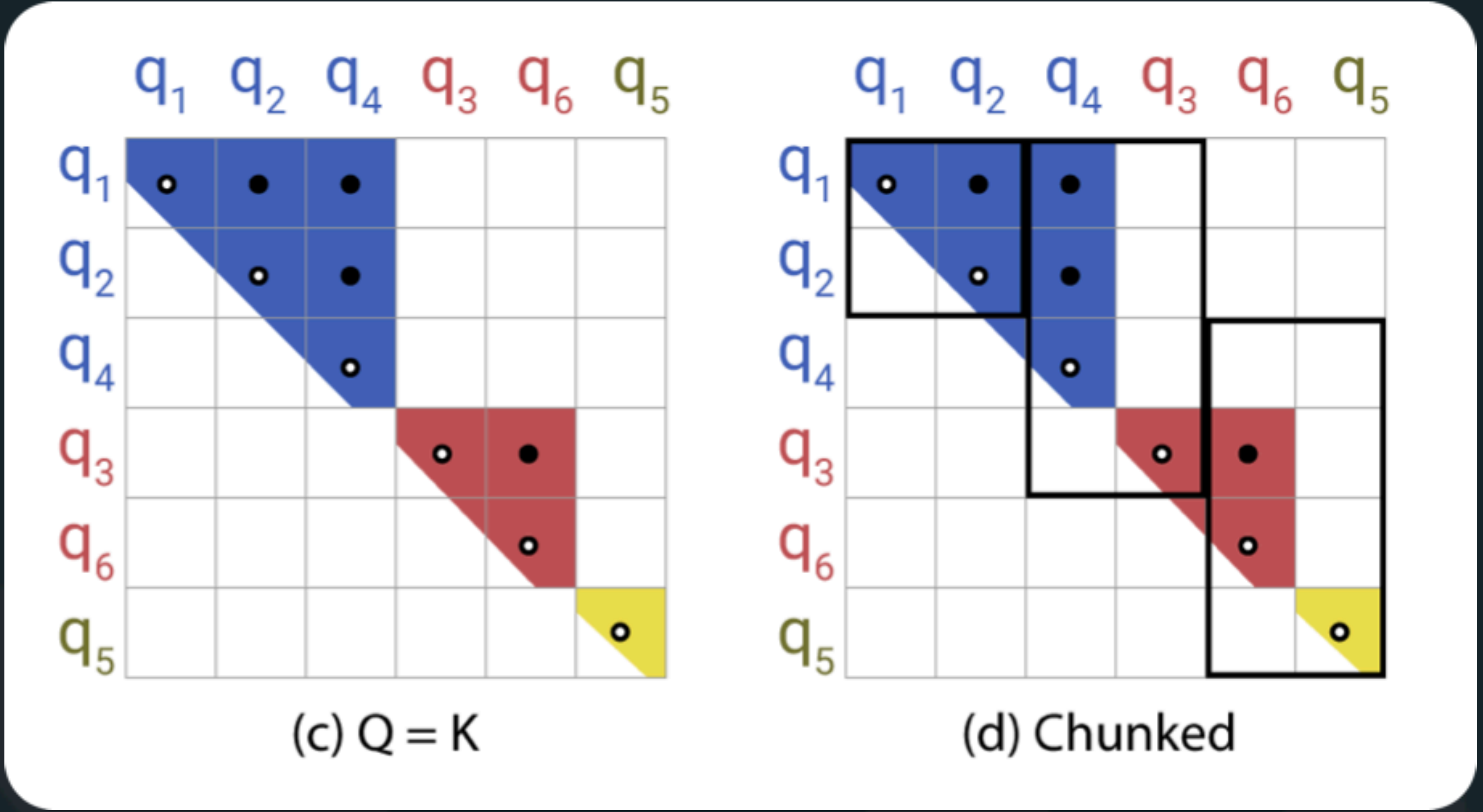
- **Problem: Batch process**

  Buckets tend to be uneven in size.

  1. A bucket may contain **many queries but no keys.**

LSH Attention

$n_{buckets}$:	3	# colors
l :	6	# queries
m :	4	# queries in 1 chunk (square)
bucket size$_{average}$:	2	# chunks in 1 rectangle



(c) $Q = K$

1. Ensure $h(k_j) = h(q_j)$ by setting $k_j = \dfrac{q_j}{||q_j||}$.

2. Sort the queries by bucket number, within each bucket, by seq. position.

(d) Chunked

1. Defines a permutation where $i \mapsto s_i$ after sorting.

2. Chunks of m consecutive queries $\tilde{\mathcal{P}}_i = \{j : \lfloor \frac{s_i}{m} \rfloor - 1 \leq \lfloor \frac{s_j}{m} \rfloor \leq \lfloor \frac{s_i}{m} \rfloor \}$. (j: in previous chunk and current chunk)
   If $\max_i |\mathcal{P}_i| < m$, then $\mathcal{P}_i \subseteq \tilde{\mathcal{P}}_i$ (where $\mathcal{P}_i = \{j : j \leq i\}$). (If the # chunk bigger then sequence length, then $\mathcal{P}_i = \tilde{\mathcal{P}}_i$)

3. In practice they set $m = \dfrac{2l}{n_{buckets}}$ (l is sequence length), the average bucket size is $\dfrac{l}{n_{buckets}}$.

4. Assume the probability of a bucket growing to twice that size is sufficiently low.

LSH Attention

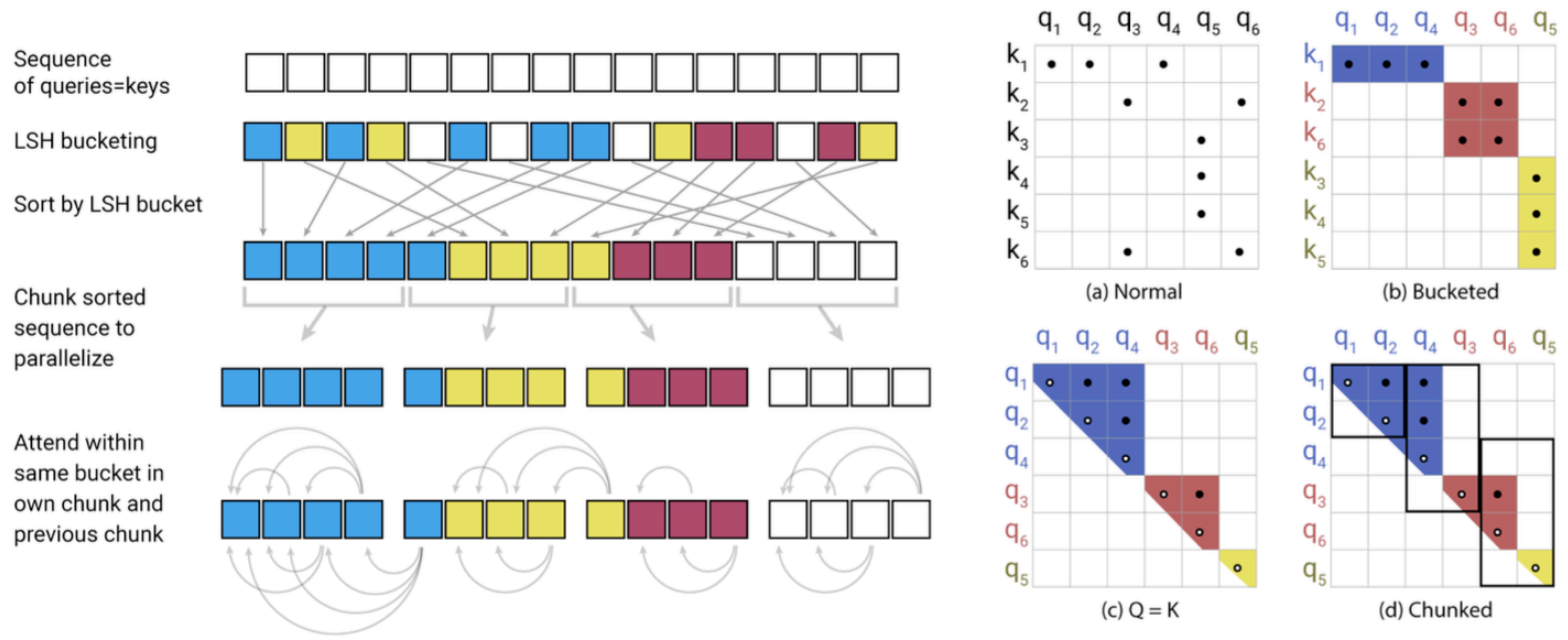


Figure 2: Simplified depiction of LSH Attention showing the hash-bucketing, sorting, and chunking steps and the resulting causal attentions. (a-d) Attention matrices for these varieties of attention.

Analysis on a synthetic task

- **Problem: Misclassification**

  There is always a small probability that similar items fall in different buckets.

- **Target: Duplicate a sequence of symbols**

  Each training & testing example has the form $0w0w$, where $w \in \{1, \ldots, N\}^*$ is a sequence of symbols ranging from 1 to N (use $N = 127$ in experiments)

  Example (w of length 3): [0, 19, 113, 72, 0, 19, 113, 72]

- **Process**

  Train a LM(predict the next symbol given all the previous ones) on examples form where each w of length 511 (so the whole input $0w0w$ is of length 1024).

Analysis on a synthetic task

- **Model structure & training parameters:**

Use a 1-layer Transformer with $d_{model} = d_{ff} = 256$, and 4 heads.

150K steps in 4 settings: Full-Attn, LSH-Attn ($n_{round} = 1$), LSH-Attn ($n_{round} = 2$), LSH-Attn ($n_{round} = 4$)

Table 2: Accuracies on the duplication task of a 1-layer Transformer model with full attention and with locality-sensitive hashing attention using different number of parallel hashes.

Train \ Eval	Full Attention	LSH-8	LSH-4	LSH-2	LSH-1
Full Attention	100%	94.8%	92.5%	76.9%	52.5%
LSH-4	0.8%	100%	99.9%	99.4%	91.9%
LSH-2	0.8%	100%	99.9%	98.1%	86.8%
LSH-1	0.8%	99.9%	99.6%	94.8%	77.9%

Causal masking in LSH Attention

- **Causal masking for shared-QK attention**

  In at Transformer decoder, masking $m(j, \mathcal{P}_i)$ is used to prevent positions from attending into the future.

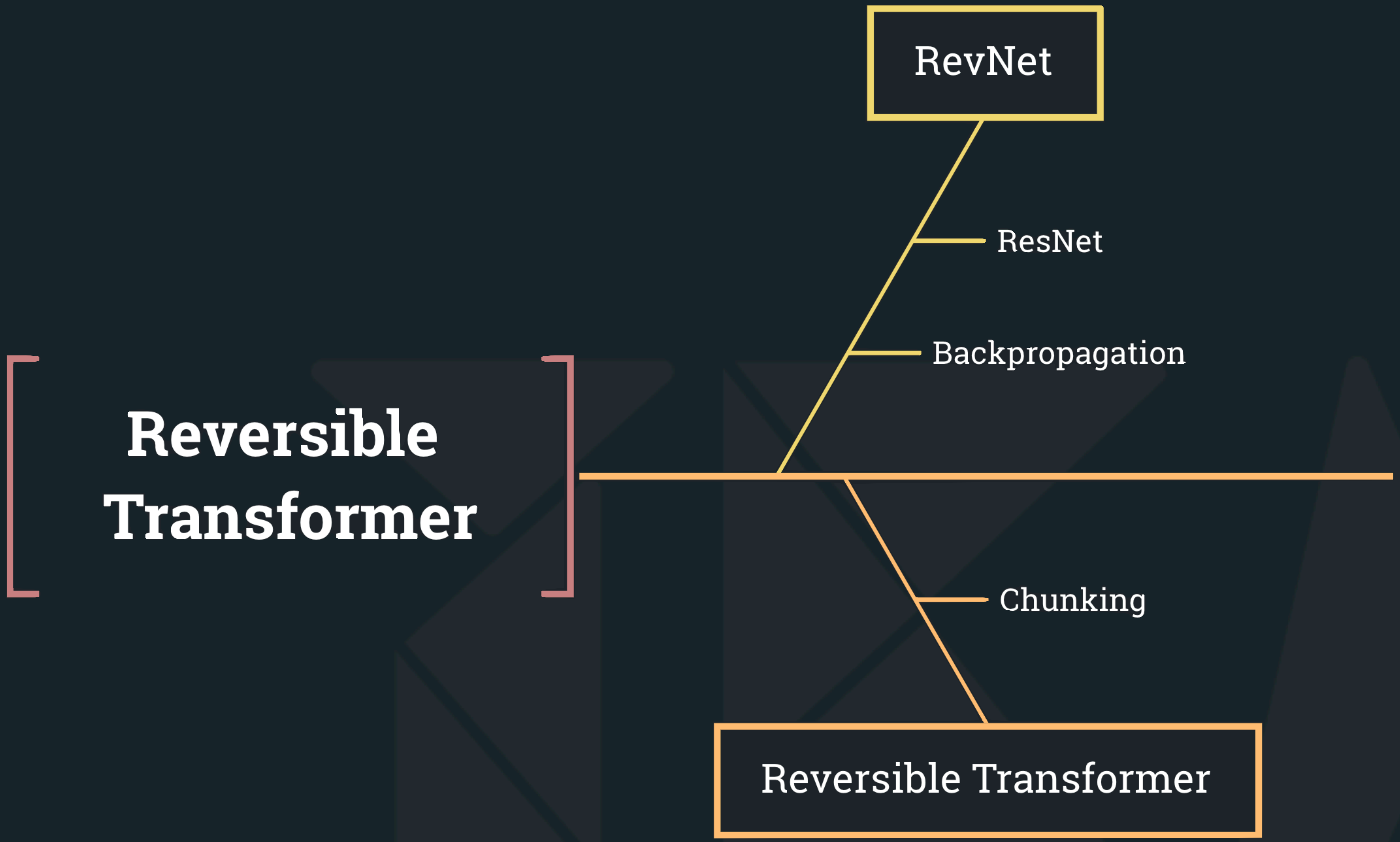
  1. Associate every query/key vector with a position index,
  2. re-order the position indices using the same permutations used to sort the query/key vectors,
  3. and then use a comparison operation to compute the mask.

- **Problem: Attend to itself**

  Modify the masking to forbid a token from attending to itself, except in situations where a token has no other valid attention targets

Reversible Transformer Part

Content



[4] Gomez, Aidan N., et al. "The reversible residual network: Backpropagation without storing activations." *Advances in neural information processing systems* 30 (2017).

ResNet brief introduction

- **Problems:**

  1. Gradient Explosion/Vanishing

  2. Degeneration:

     e.g. If the best layer number is 18, but we designed 34 layers for this problem. Then another 16 layers must learn the **Identity Mapping (if f is a IM then $x_{out} = f(x) = x_{in}$)**, but the model can't learn the perfect IM generally. Therefore the redundant 16 layers would dropping down the entire model.

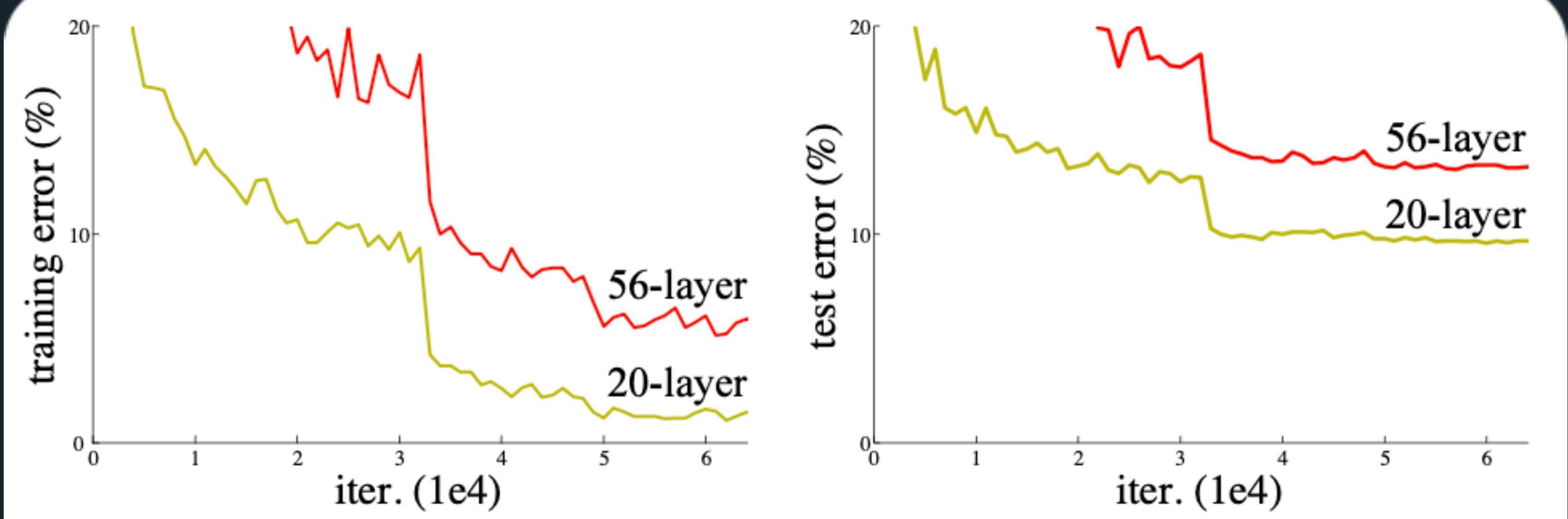
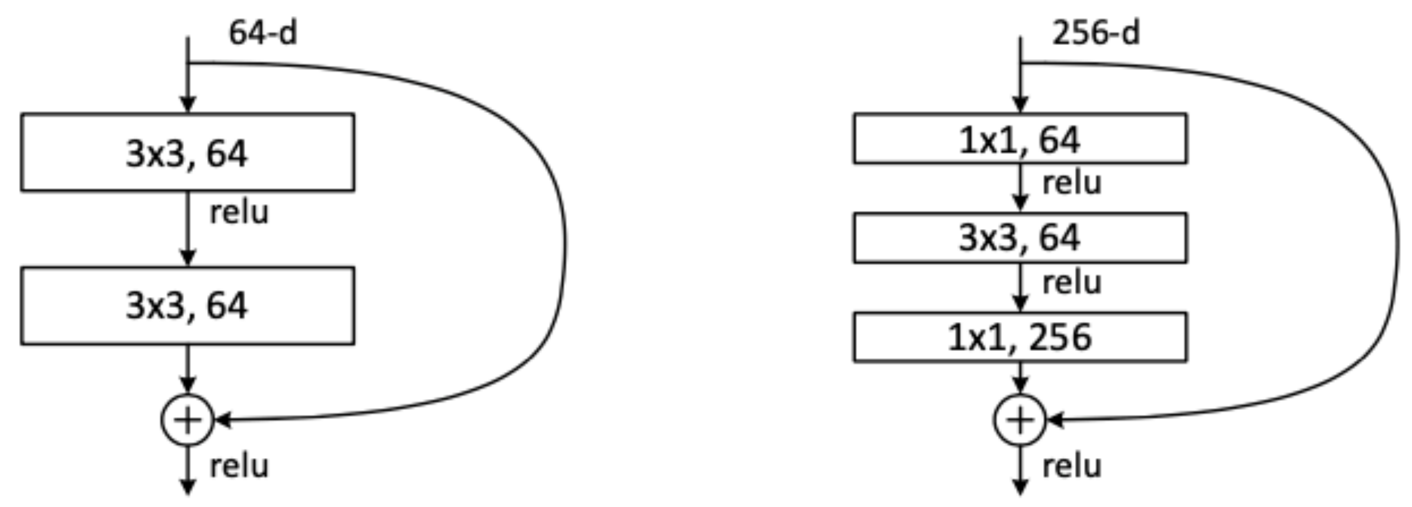


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

- **Identity mapping by shortcuts:**

$$\mathbf{y} = F(\mathbf{x}, \{W_i\}) + \mathbf{x}$$



Here $\mathbf{x}$ and $\mathbf{y}$ are the input and output vectors of the layers considered. The function $F(\mathbf{x}, \{W_i\})$ represents the residual mapping to be learned.

[5] Tai, Ying, Jian Yang, and Xiaoming Liu. "Image super-resolution via deep recursive residual network." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

Model Structure of RevNet

- **In classic backpropagation parameters update**

$$\bar{v}_i = \sum_{j \in \text{Child(i)}} \left(\frac{\partial f_j}{\partial v_i}\right)^T \bar{v}_j, \quad \text{where } \bar{v}_i \text{ denotes the total derivative}$$

- **Architecture of Reversible Residual Block**

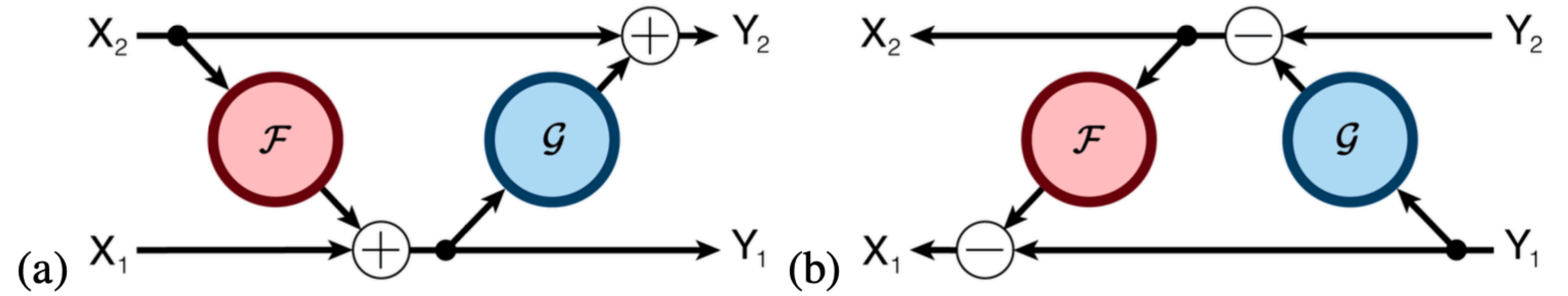


Figure 2: **(a)** the forward, and **(b)** the reverse computations of a residual block, as in Equation 8.

$$y_1 = x_1 + F(x_2) \qquad x_2 = y_2 - G(y_1)$$

$$y_2 = x_2 + F(y_1) \qquad x_1 = y_1 + F(x_2)$$

Dinh, Laurent, David Krueger, and Yoshua Bengio. "Nice: Non-linear independent components estimation." *arXiv preprint arXiv:1410.8516* (2014) ICLR 2015.

Reversible Residual Block Backprop Algorithm

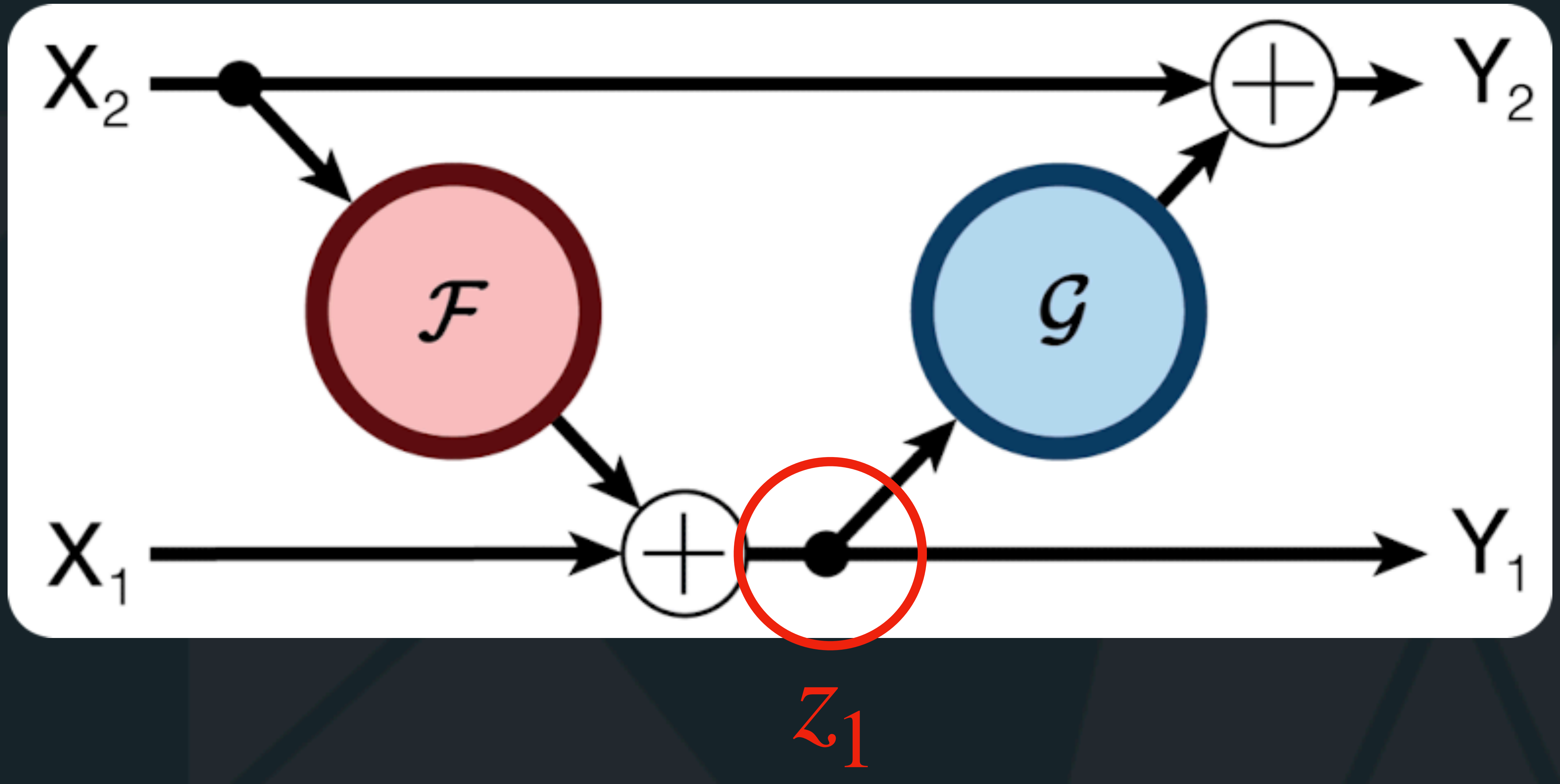
- **Algorithm: Block Reverse**

function BlockReverse$((y_1, y_2), (\bar{y}_1, \bar{y}_2))$

1. $z_1 \leftarrow y_1$
2. $x_2 \leftarrow y_2 - G(z_1)$
3. $x_1 \leftarrow z_1 - F(x_2)$
4. $\bar{z}_1 \leftarrow \bar{y}_1 + (\frac{\partial G}{\partial z_1})^\intercal\ \bar{y}_2$
5. $\bar{x}_2 \leftarrow \bar{y}_2 + (\frac{\partial F}{\partial x_2})^\intercal\ \bar{z}_1$
6. $\bar{x}_1 \leftarrow \bar{z}_1$
7. $\bar{w}_F \leftarrow (\frac{\partial F}{\partial w_F})^\intercal\ \bar{z}_1$
8. $\bar{w}_G \leftarrow (\frac{\partial G}{\partial w_G})^\intercal\ \bar{y}_2$

return $(x_1, x_2), (\bar{x}_1, \bar{x}_2), (\bar{w}_F, \bar{w}_G)$

end function BlockReverse



Reversible Transformer

- **Original Architecture of Reversible Residual Block**

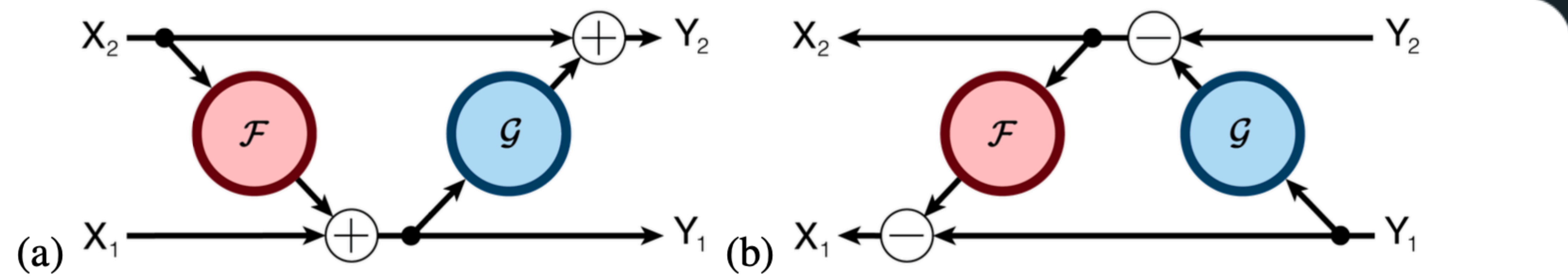
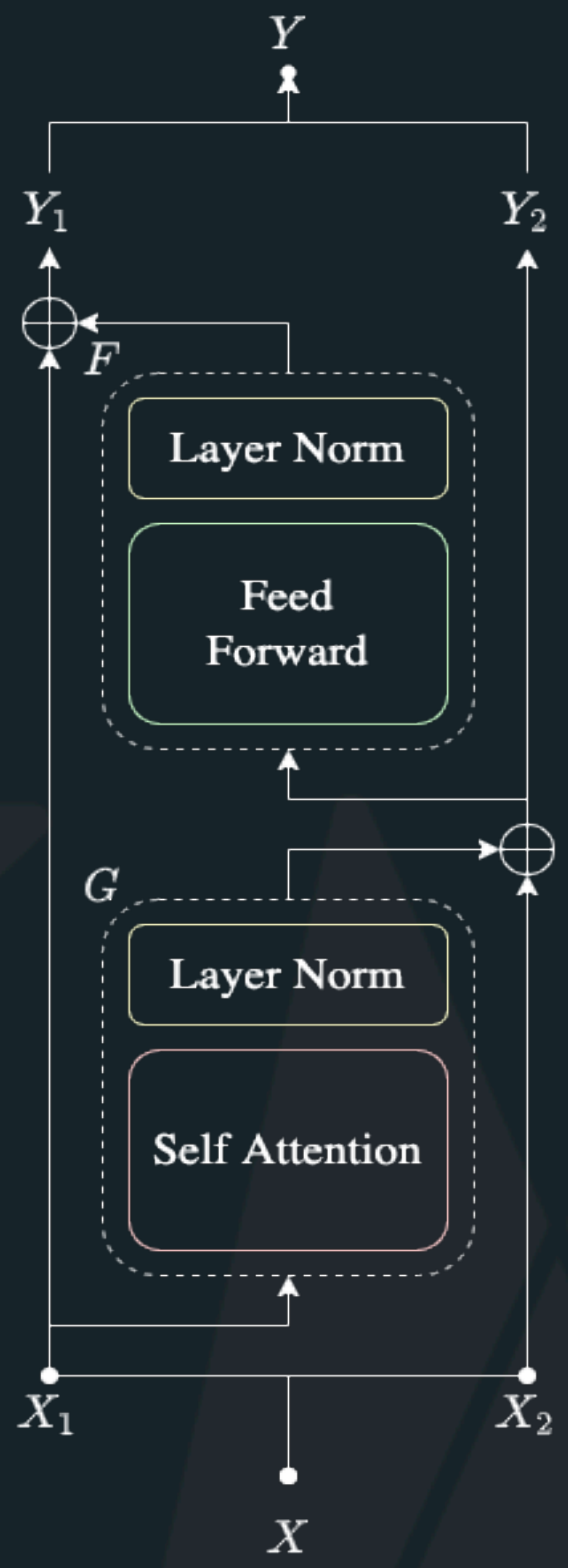


Figure 2: **(a)** the forward, and **(b)** the reverse computations of a residual block, as in Equation 8.

$$y_1 = x_1 + F(x_2) \qquad x_2 = y_2 - G(y_1)$$
$$y_2 = x_2 + F(y_1) \qquad x_1 = y_1 + F(x_2)$$

- **Reversible Transformer**

$$\mathrm{Y}_1 = \mathrm{X}_1 + \mathrm{Attention}(\mathrm{X}_2) \qquad \mathrm{Y}_2 = \mathrm{X}_2 + \mathrm{FeedForward}(\mathrm{Y}_1)$$



Complexity Analysis

b :	batch size
l :	sequence length
d_{ff} :	depth of FF
d_{model} :	depth of model
n_h :	# head
n_l :	# layer
n_r :	# LSH round
n_c :	# LSH chunk

Table 3: Memory and time complexity of Transformer variants. We write d_{model} and d_{ff} for model depth and assume $d_{ff} \geq d_{model}$; b stands for batch size, l for length, n_l for the number of layers. We assume $n_c = l/32$ so $4l/n_c = 128$ and we write $c = 128^2$.

Model Type	Memory Complexity	Time Complexity
Transformer	$\max(bld_{ff}, bn_h l^2)n_l$	$(bld_{ff} + bn_h l^2)n_l$
Reversible Transformer	$\max(bld_{ff}, bn_h l^2)$	$(bn_h ld_{ff} + bn_h l^2)n_l$
Chunked Reversible Transformer	$\max(bld_{model}, bn_h l^2)$	$(bn_h ld_{ff} + bn_h l^2)n_l$
LSH Transformer	$\max(bld_{ff}, bn_h ln_r c)n_l$	$(bld_{ff} + bn_h n_r lc)n_l$
Reformer	$\max(bld_{model}, bn_h ln_r c)$	$(bld_{ff} + bn_h n_r lc)n_l$

Experiments - Effect of Share-QK & Reversible Layers

Effect of Share-QK

- Set $k_j = \dfrac{q_j}{\| q_j \|}$
- Prevents attending to itself.
- For enwik8 share-QK appears to train slightly faster.
- Without sacrificing accuracy.

Effect of Reversible Layers

- Memory saving in Reversible Transformer don't come at the expense of accuracy.

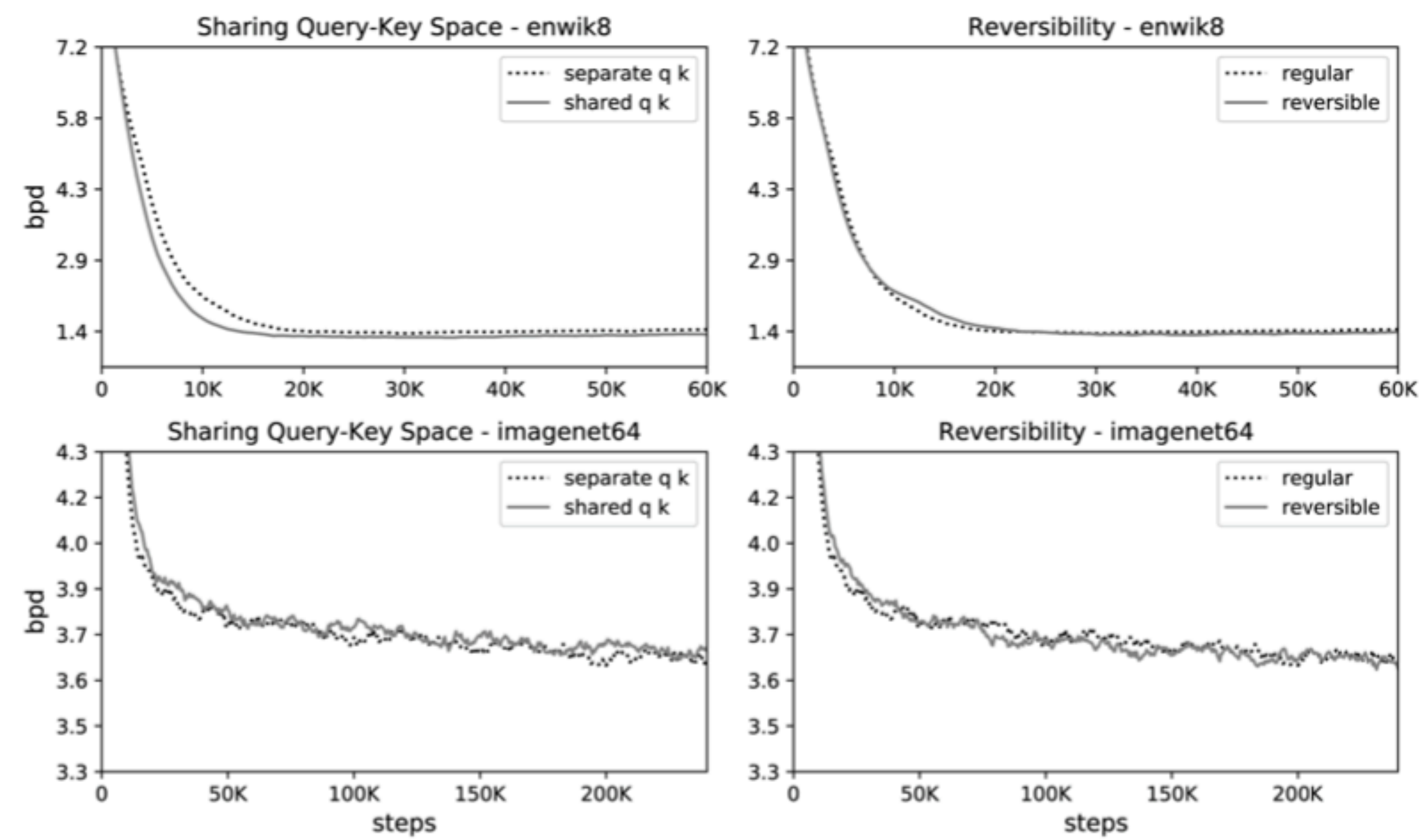


Figure 3: Effect of shared query-key space (left) and reversibility (right) on performance on enwik8 and imagenet64 training. The curves show bits per dim on held-out data.

Experiments - Reversible layers in machine translation.

BLEU scores on newstest2014 for WMT EnGe

- Without LSH Attention
- Typical LSH Attention configuration uses chunks of 128 tokens after hashing and sorting, whereas the examples in the WMT14 test set are all shorter than 128 tokens.

Table 4: BLEU scores on newstest2014 for WMT English-German (EnDe). We additionally report detokenized BLEU scores as computed by sacreBLEU (Post, 2018).

Model	BLEU	*sacreBLEU* *Uncased*[3]	*sacreBLEU* *Cased*[4]
Vaswani et al. (2017), base model	27.3		
Vaswani et al. (2017), big	28.4		
Ott et al. (2018), big	29.3		
Reversible Transformer (base, 100K steps)	27.6	*27.4*	*26.9*
Reversible Transformer (base, 500K steps, no weight sharing)	28.0	*27.9*	*27.4*
Reversible Transformer (big, 300K steps, no weight sharing)	29.1	*28.9*	*28.4*

Experiments - LSH Attention in Transformer & Large Reformer Models

LSH Attention in Transformer

- At $n_{rounds} = 8$, almost matches full attention.
- LSH attention speed remains flat.

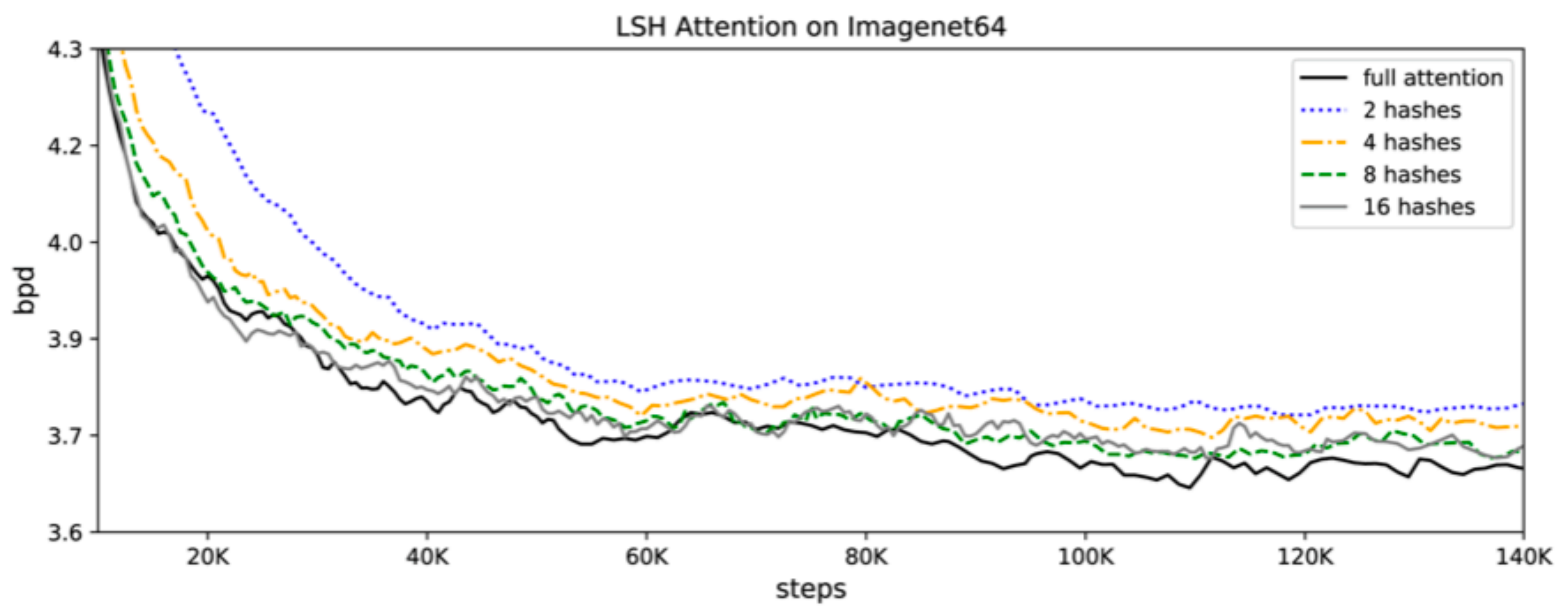


Figure 4: LSH attention performance as a function of hashing rounds on imagenet64.

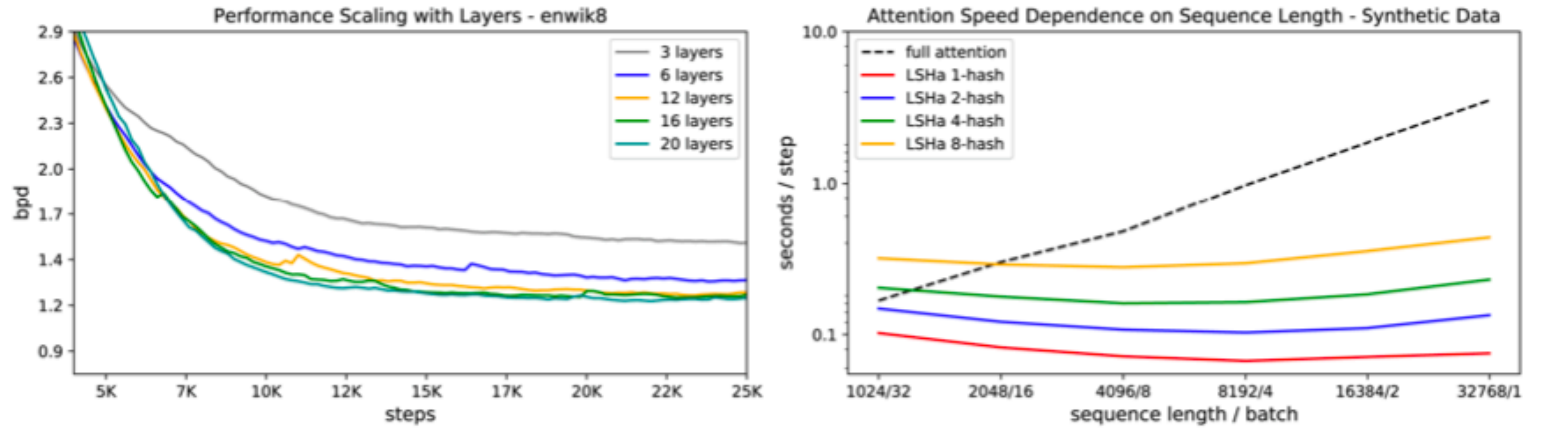


Figure 5: Left: LSH attention performance as a function of number of layers on enwik8. Right: Speed of attention evaluation as a function of input length for full- and LSH- attention.

Large Reformer Models

- 12-Layer on enwik8 trained for 20K steps with a dropout rate of 0.1 achieves 1.18 bpd on test set.
- With further tuning they reached 1.05 bpd.

SWOT Analysis

Strength	Weakness
• Longer sequence length. • More flexible depth of model. • Portability.	• Low adjustability of LSH Attention.
Opportunity	**Threat**
• Bring the power of Transformer models to other domains like time-series forecasting, music, image and video generation.	• Using RevNet would consume more 50% time to do the Block Reverse Algorithm.