

# LLM Sys

## Pre-trained LLMs

Lei Li



Language  
Technologies  
Institute

**Carnegie Mellon University**  
School of Computer Science

# Recap

- Sequence-to-sequence encoder-decoder framework for conditional generation, including Machine Translation
- Key components in Transformer
  - Positional Embedding (to distinguish tokens at different pos)
  - Multihead attention
  - Residual connection
  - Layer norm

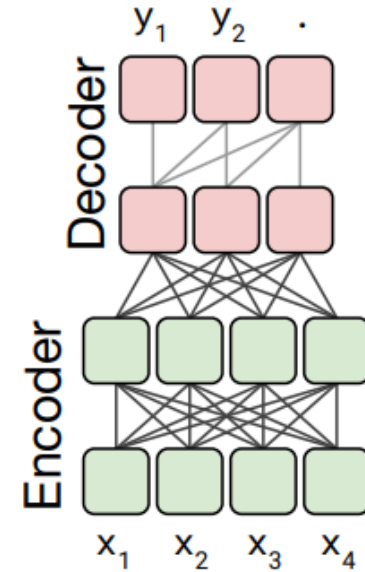
# Today's Topic

- Pretrained Encoder-Decoder LLM: T5
- Pretrained Decoder-Only LLMs: LLaMA
- GPT3

# T5

- Model Architecture

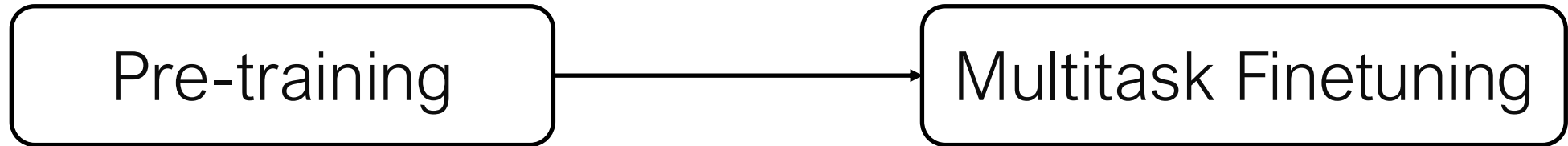
- Standard encoder-decoder Transformer
- Decoding: beam search
  - Beam width=4, length penalty=0.6



- Model Size

- T5-base: 220 million parameters
  - 12 blocks,  $d_{\text{ff}} = 3072$ ,  $d_{\text{kv}} = 64$ , 12-headed attention,  $d_{\text{model}} = 768$
- T5-3B
  - 24 blocks,  $d_{\text{model}} = 1024$ ,  $d_{\text{kv}} = 128$ ,  $d_{\text{ff}} = 16384$ , 32-headed attention
- T5-11B
  - 24 blocks,  $d_{\text{model}} = 1024$ ,  $d_{\text{kv}} = 128$ ,  $d_{\text{ff}} = 65536$ , 128-headed attention

# Training



- Pretraining: C4: filtered English corpus from Common Crawl
  - 750GB, still used often today
- Supervised fine-tuning
  - Language understanding/Text classification: GLUE/SuperGLUE
  - Summarization: CNN/Daily mail corpus
  - Question answering: SQuAD
  - Translation: WMT English to German, French, Romanian

# T5 Pre-training: Recover randomly corrupted spans

Standard next token cross entropy loss, plus ... Cloze-style QA  
15% of text are corrupted

Original text

Thank you ~~for~~ ~~inviting~~ me to your party ~~last~~ week.

Inputs

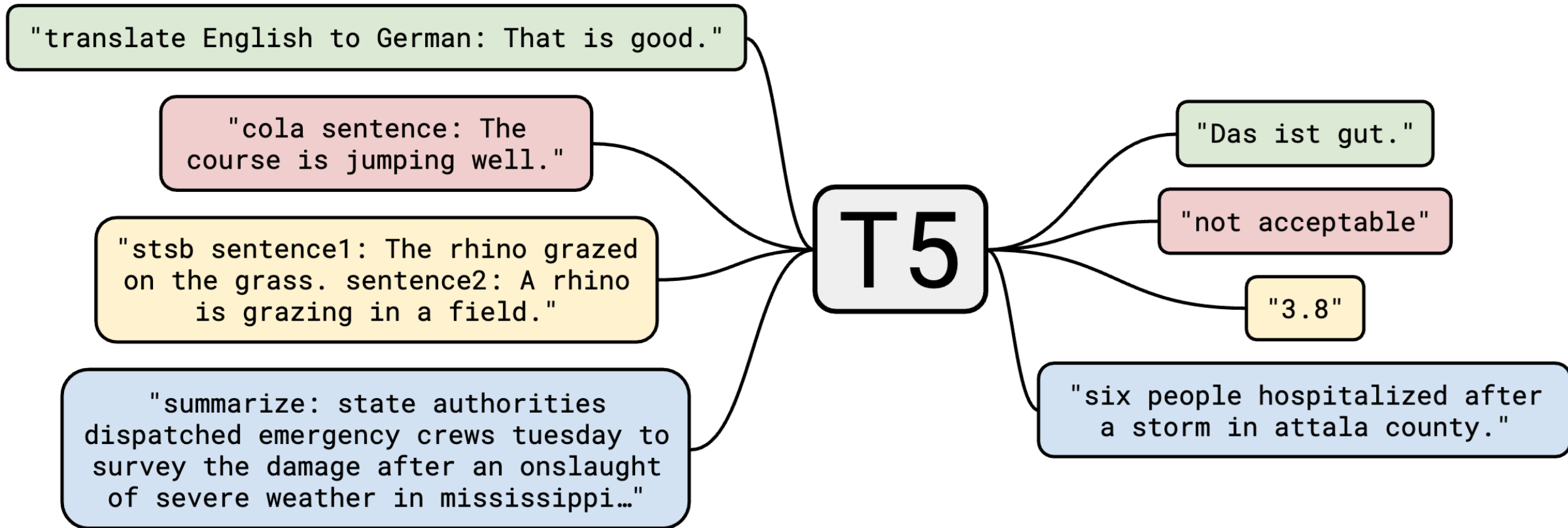
Thank you <X> me to your party <Y> week.

Targets

<X> for inviting <Y> last <Z>

Pre-train for 0.5 million steps on a batch size of 128 sequences of length 512.  
packing multiple seqs 65k tokens per batch, result in 34B trained tokens.

# T5 Multitask SFT with Task Instruction



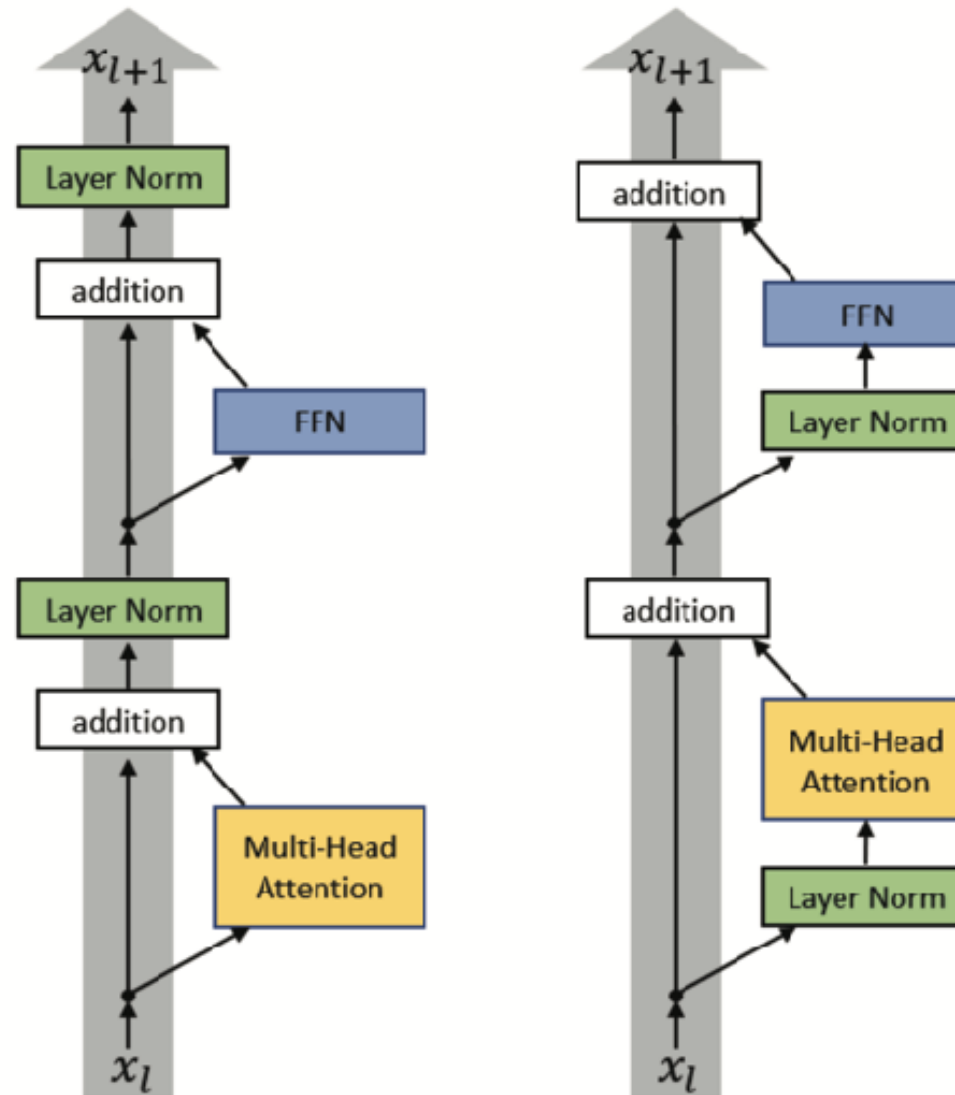
Unified format to put task instructions as natural language in the input, enables transfer to new tasks. with more instruction tuning → T0, Flan-T5<sub>7</sub>

# LLaMA

- Model Architecture:
  - Based on Transformer decoder-only, with a few improvements.
  - Pre-normalization [GPT3]
  - SwiGLU activation function [PaLM]: Swish-Gated Linear Unit
  - Rotary Embeddings [RoFormer]

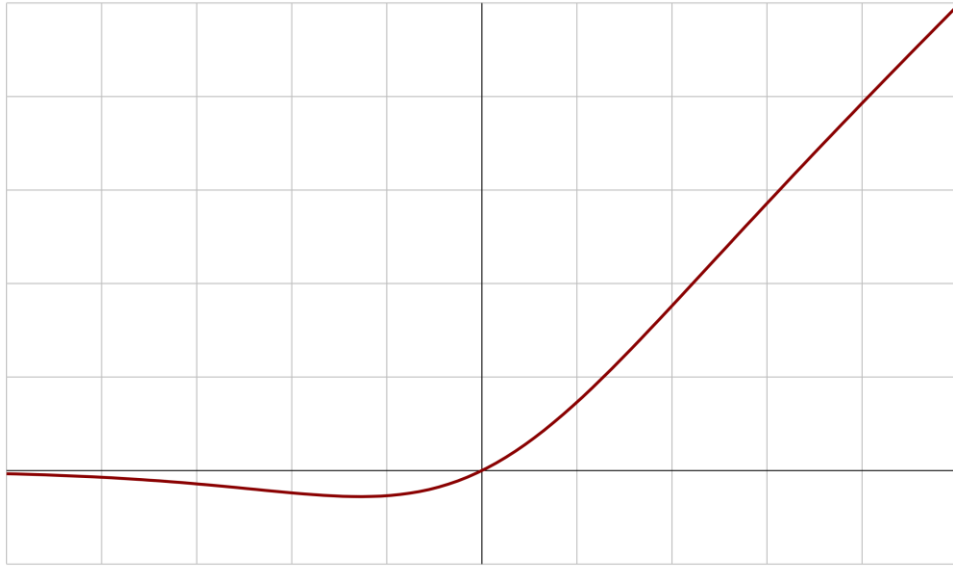


# pre Layer Normalization



# FFN with SwiGLU

Swish activation



$$swish(x) = x \sigma(\beta x)$$

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

FFN with ReLU

$$FFN(x) = \max(0, x \cdot W_1 + b_1) \cdot W_2 + b_2$$

dim=4d

FFN with SwiGLU

$$FFN_{SwiGLU}(x) = (Swish(x \cdot W_1 + b_1) \odot (x \cdot W_2 + b_2)) \cdot W_3 + b_2$$

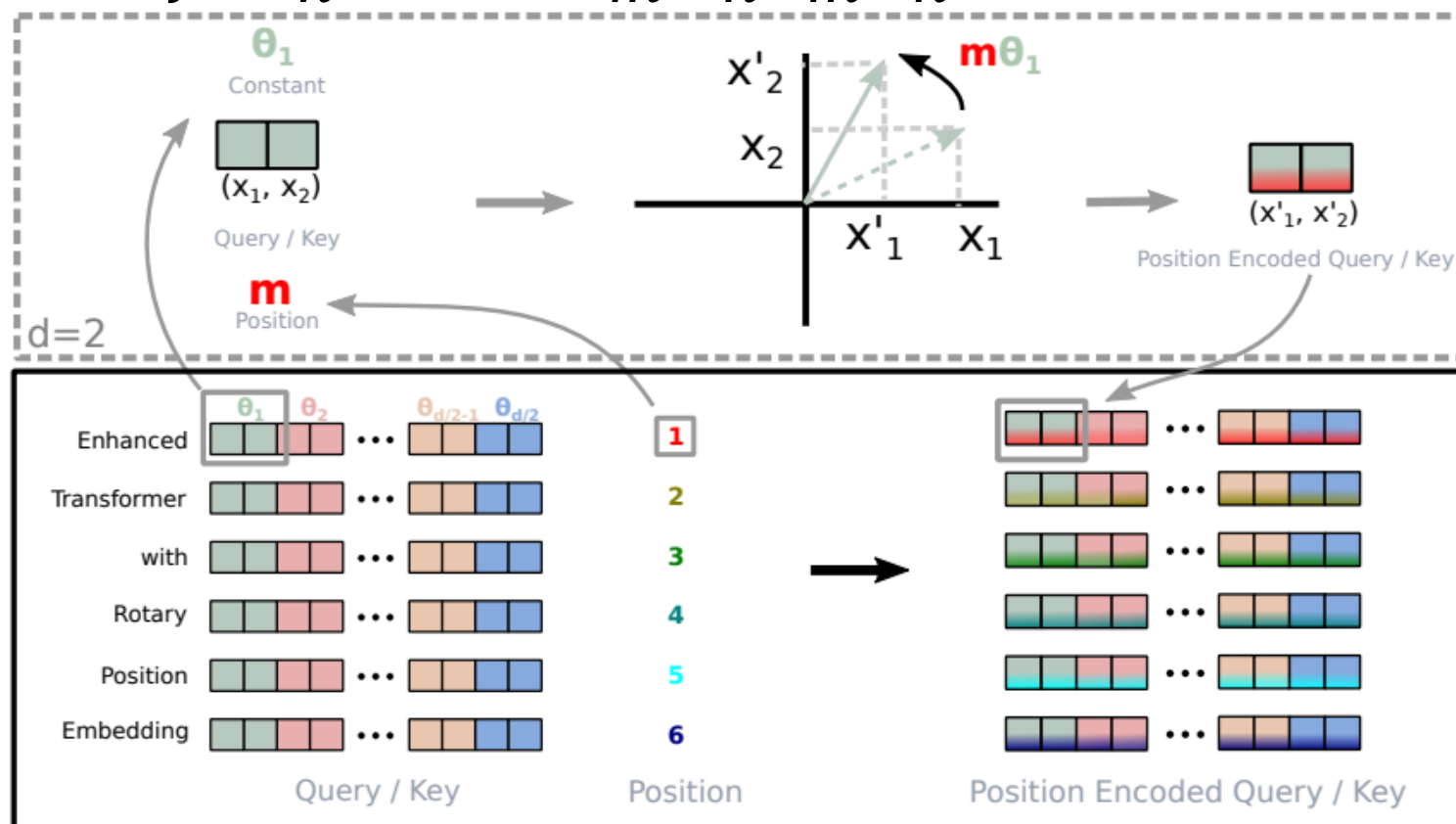
dim=2/3 \* 4d

# Rotary Embedding (RoPE)

make the attention weight depending (only) on position distance

$$f(x_m, m) = \begin{pmatrix} \cos(m\theta) & -\sin(m\theta) \\ \sin(m\theta) & \cos(m\theta) \end{pmatrix} \begin{pmatrix} x_{m,1} \\ x_{m,2} \end{pmatrix}$$

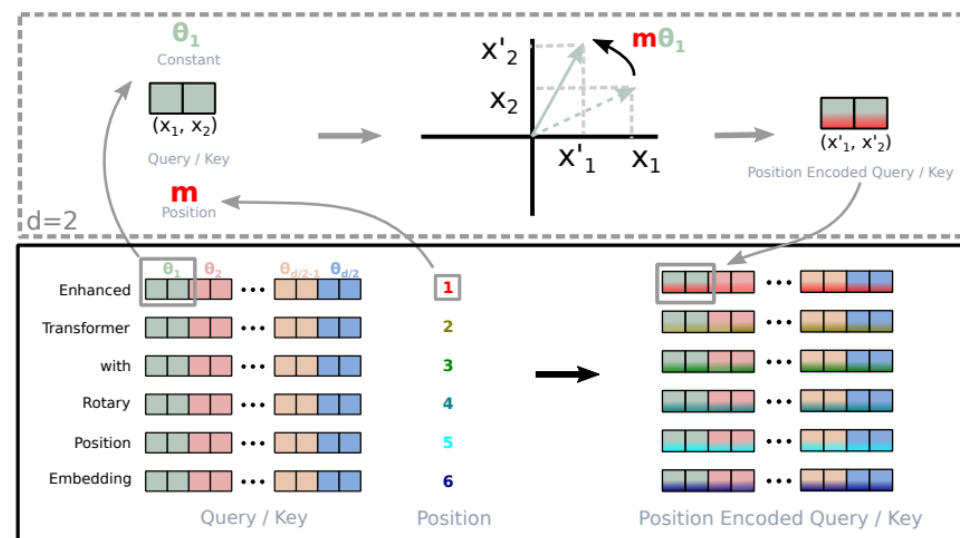
$$f(x_m, m) \cdot f(x_n, n) = x_m^T R_{n-m} x_n$$



# Rotary Embedding (RoPE)

$$f(x_m, m) = \begin{pmatrix} \cos(m\theta_1) & -\sin(m\theta_1) \\ \sin(m\theta_1) & \cos(m\theta_1) \\ & \cos(m\theta_2) & -\sin(m\theta_2) \\ & \sin(m\theta_2) & \cos(m\theta_2) \end{pmatrix} \begin{pmatrix} x_{m,1} \\ x_{m,2} \\ x_{m,3} \\ x_{m,4} \end{pmatrix}$$

$$f(x_m, m)^T \cdot f(x_n, n) = x_m^T R_{n-m} x_n$$



# LLaMA

- Model Size

params	dimension	$n$ heads	$n$ layers	learning rate	batch size	$n$ tokens
6.7B	4096	32	32	$3.0e^{-4}$	4M	1.0T
13.0B	5120	40	40	$3.0e^{-4}$	4M	1.0T
32.5B	6656	52	60	$1.5e^{-4}$	4M	1.4T
65.2B	8192	64	80	$1.5e^{-4}$	4M	1.4T

# LLaMA

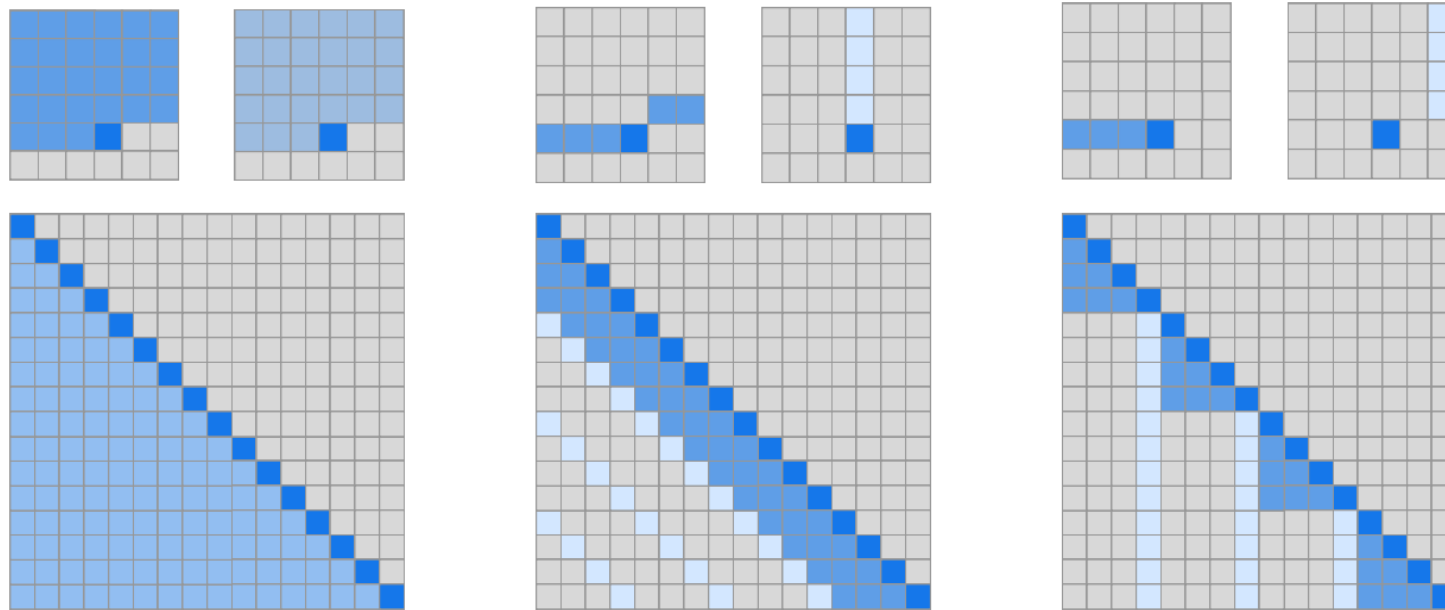
- Training Strategy
  - Trained with the standard language modeling loss function: the average log probability of all tokens without label smoothing
  - Auxiliary loss to encourage the softmax normalizer to be close to 0
- Pre-training Details
  - Using only open-source data

Dataset	Sampling prop.	Epochs	Disk size
CommonCrawl	67.0%	1.10	3.3 TB
C4	15.0%	1.06	783 GB
Github	4.5%	0.64	328 GB
Wikipedia	4.5%	2.45	83 GB
Books	4.5%	2.23	85 GB
ArXiv	2.5%	1.06	92 GB
StackExchange	2.0%	1.03	78 GB

# GPT3

- Model Architecture

- Based on the standard Transformer architecture
- With modified initialization, pre-normalization, and reversible tokenization
- Alternating dense and locally banded **sparse attention** patterns



(a) Transformer

(b) Sparse Transformer (strided)

(c) Sparse Transformer (fixed)

# GPT3

- Model Size

Model Name	$n_{\text{params}}$	$n_{\text{layers}}$	$d_{\text{model}}$	$n_{\text{heads}}$	$d_{\text{head}}$	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	$6.0 \times 10^{-4}$
GPT-3 Medium	350M	24	1024	16	64	0.5M	$3.0 \times 10^{-4}$
GPT-3 Large	760M	24	1536	16	96	0.5M	$2.5 \times 10^{-4}$
GPT-3 XL	1.3B	24	2048	24	128	1M	$2.0 \times 10^{-4}$
GPT-3 2.7B	2.7B	32	2560	32	80	1M	$1.6 \times 10^{-4}$
GPT-3 6.7B	6.7B	32	4096	32	128	2M	$1.2 \times 10^{-4}$
GPT-3 13B	13.0B	40	5140	40	128	2M	$1.0 \times 10^{-4}$
GPT-3 175B or “GPT-3”	175.0B	96	12288	96	128	3.2M	$0.6 \times 10^{-4}$

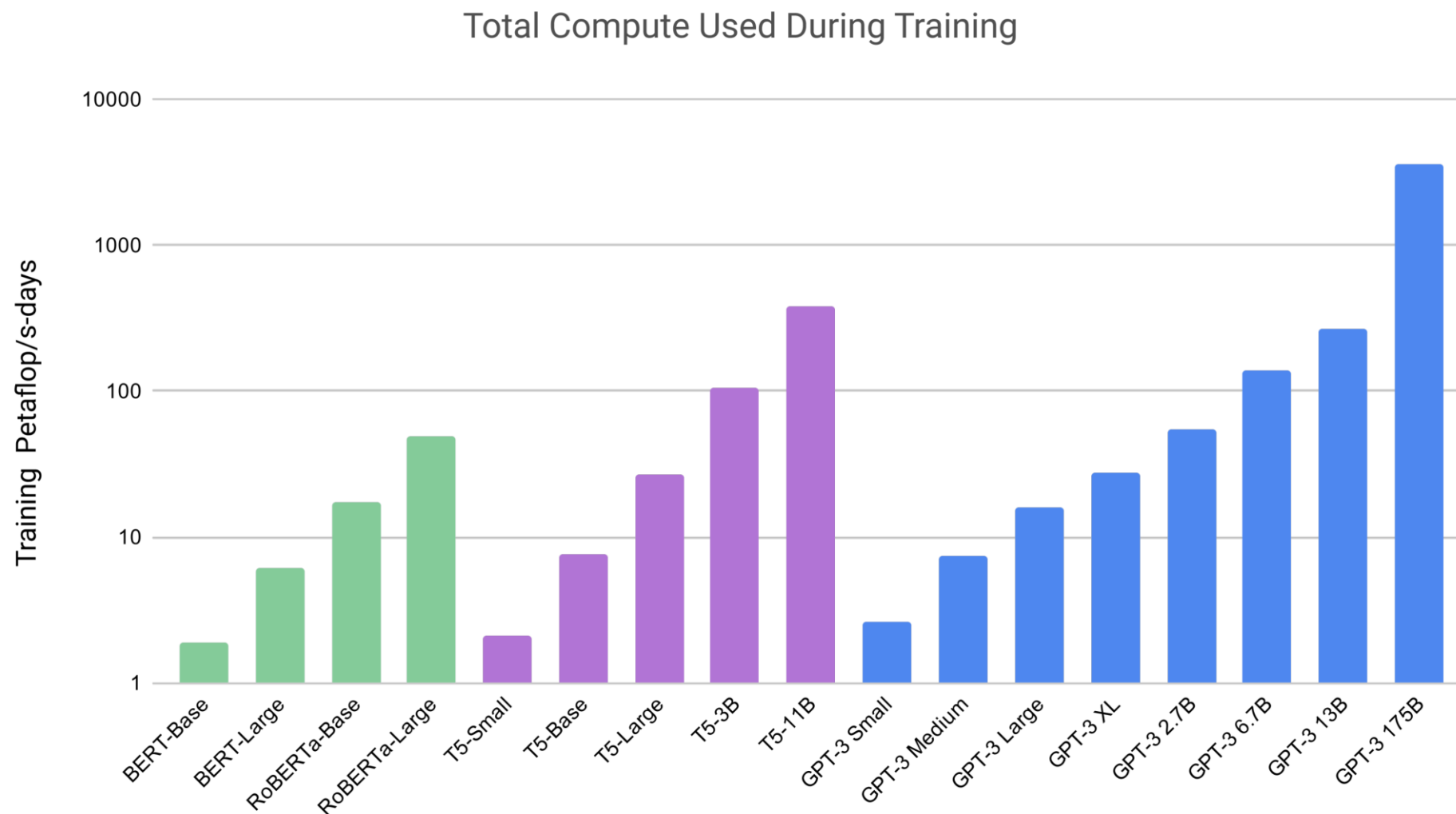


# GPT3

- Training Strategy
  - Unsupervised Pre-training
- Training Details

Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

# Computation



# Quiz

<https://canvas.cmu.edu/courses/51352/quizzes/155254>

# Summary – Key Ideas in Modern Pre-trained LLMs

- Pretraining: Mask-labeled recovery of random spans + next token prediction
- Multitask supervised fine-tuning with instruction templates
  - T5, InsturctGPT
- Relative position: Rotary positional embedding
- Smooth activation: SwishGLU
- Sparse attention

# HW3

[https://lmsystem.github.io/lmsystemhomework/assignment\\_3/](https://lmsystem.github.io/lmsystemhomework/assignment_3/)

# Reading for Next Class

- Neural Machine Translation of Rare Words with Subword Units. Sennrich et al. 2016.
- SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. Kudo and Richardson. 2018