



# *LoRA: Low-Rank Adaptation of Large Language Models*

Edward Hu\*   Yelong Shen\*   Phillip Wallis  
Zeyuan Allen-Zhu   Yuanzhi Li   Shean Wang   Weizhu Chen  
Microsoft Corporation  
{edwardhu, yeshe, phwallis, zeyuana, yuanzhil  
swang, wzchen}@microsoft.com

汇报人: 蒋明忠

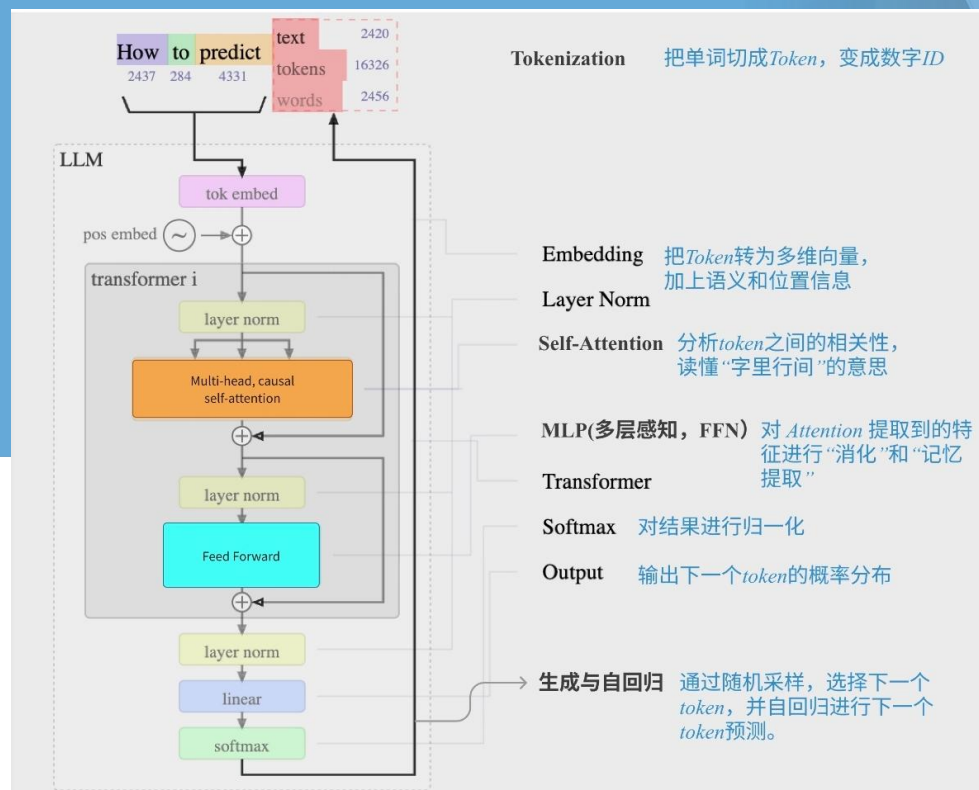
时间: 2026.04



# Background



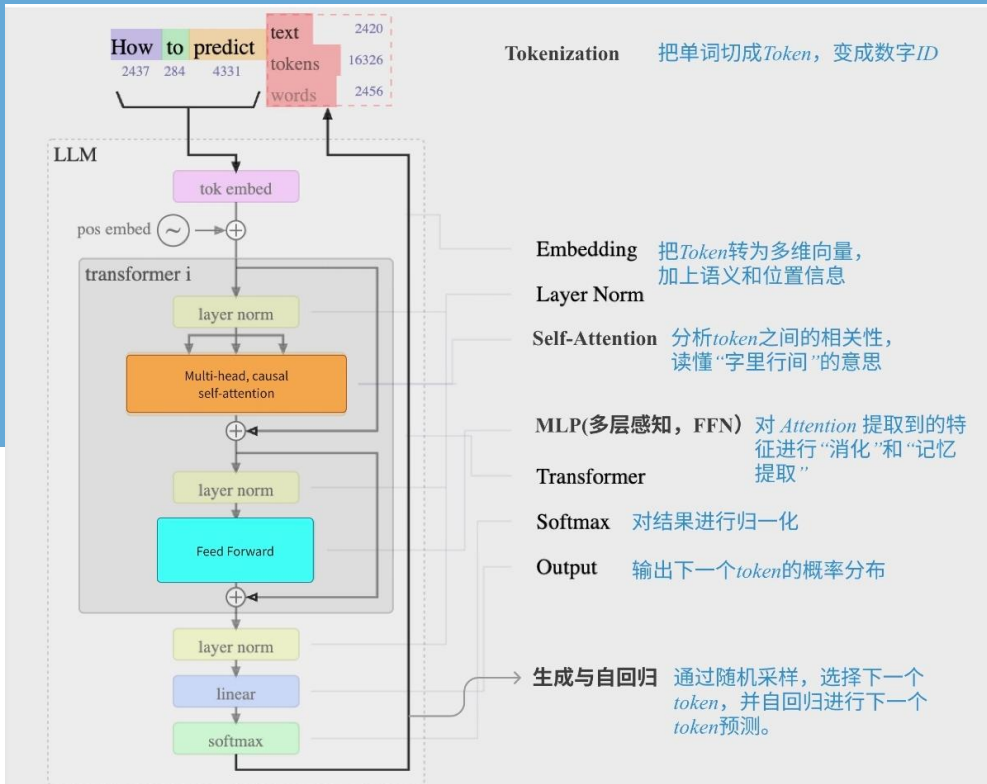
自然语言处理的主流范式包括对**一般领域数据的大规模预训练**和对**特定任务或领域的适应**。当我们预训练更大的模型时，重新训练**所有模型参数**的传统微调变得不太可行。以GPT-3-175B为例，部署许多独立的微调模型（每个模型实例都有175B参数）是非常昂贵的。



# Background



**Transformer**是一个序列到序列的架构，它大量使用了自关注。基于Transformer的语言模型主导了NLP，在许多任务中实现了最先进的技术。GPT-3就是在大量文本上训练的大型Transformer语言模型。**GPT3-175B: 96层、hidden size 12,288、96个attention heads**



$$Q, K, V = xW_Q^T, xW_K^T, xW_V^T \quad (W \in \mathbb{R}^{d_{model} \times d_{model}})$$

$$head_i = \text{Softmax} \left( \frac{Q_i K_i^T}{\sqrt{d_{head}}} + M \right) V_i \quad (i = 1 \dots h)$$

$$\text{MHSA}(x) = \text{Concat}(\text{head}_1, \dots, \text{head}_h) W_O^T \quad (W_O \in \mathbb{R}^{d_{model} \times d_{model}})$$

$$\max_{\Phi} \sum_{(x,y) \in \mathcal{Z}} \sum_{t=1}^{|y|} \log(p_{\Phi}(y_t|x, y_{<t}))$$

# Method



我们提出了低秩自适应(Low-Rank Adaptation,LoRA),它冻结了预训练的模型权重,并将可训练的秩分解矩阵注入到Transformer体系结构的每一层,从而大大减少了下游任务的可训练参数的数量。

$$Q, K, V = xW_Q^T, xW_K^T, xW_V^T \quad (W \in \mathbb{R}^{d_{model} \times d_{model}})$$

$$head_i = \text{Softmax} \left( \frac{Q_i K_i^T}{\sqrt{d_{head}}} + M \right) V_i \quad (i = 1 \dots h)$$

$$\text{MHSA}(x) = \text{Concat}(\text{head}_1, \dots, \text{head}_h) W_O^T \quad (W_O \in \mathbb{R}^{d_{model} \times d_{model}})$$

$$h = W_0 x + \Delta W x = W_0 x + B A x$$

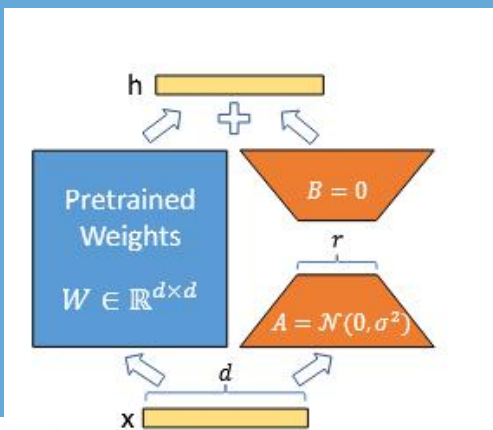
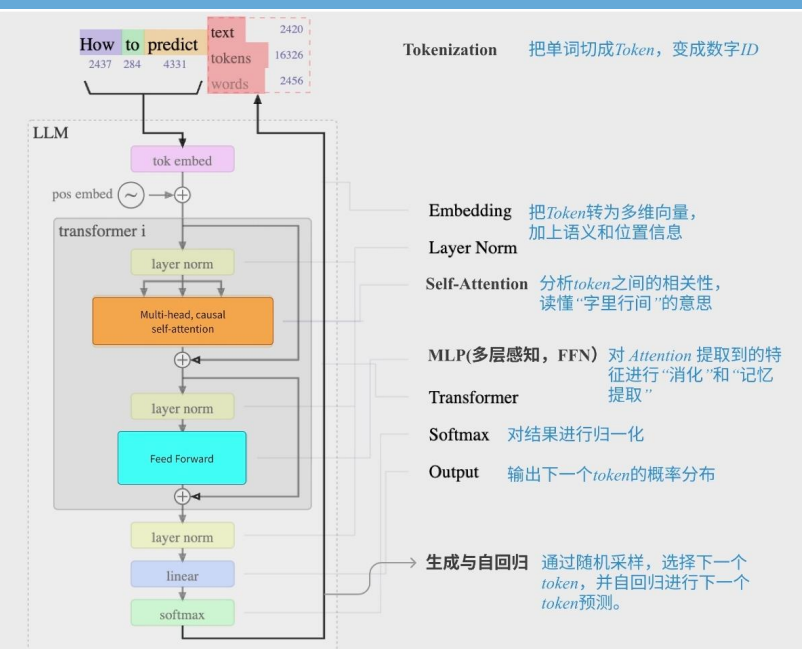


Figure 1: Our reparametrization. We only train  $A$  and  $B$ .

$$\max_{\Theta} \sum_{(x,y) \in \mathcal{Z}} \sum_{t=1}^{|y|} \log(p_{\Phi_0 + \Delta\Phi(\Theta)}(y_t | x, y_{<t}))$$

# Method



我们提出了低秩自适应(Low-Rank Adaptation,LoRA),它冻结了预训练的模型权重,并将可训练的秩分解矩阵注入到Transformer体系结构的每一层,从而大大减少了下游任务的可训练参数的数量。

GPT3-175B: 96 层、hidden size 12,288、96 个 attention heads

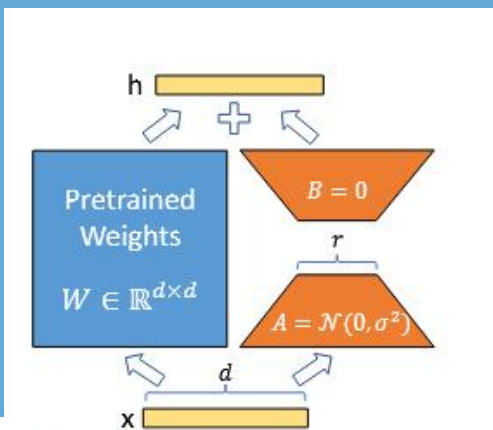
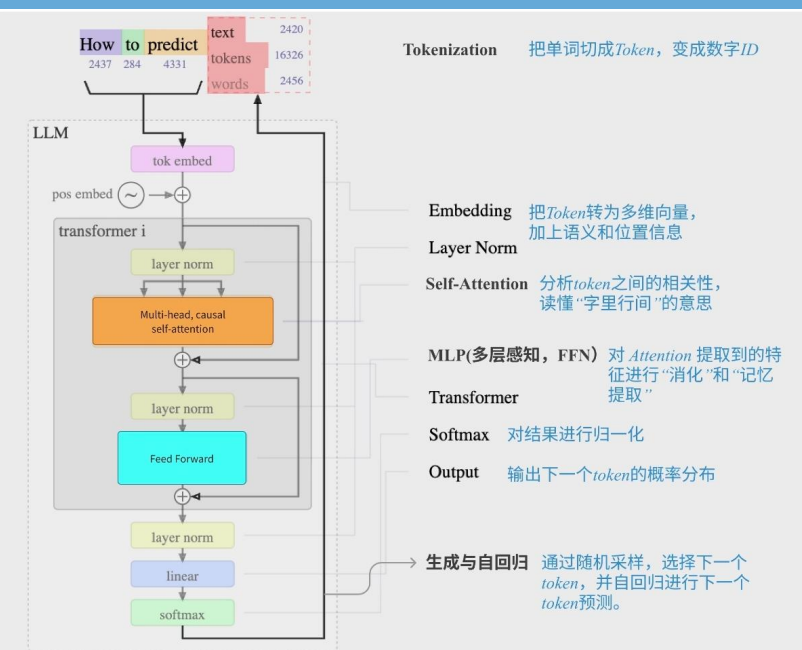


Figure 1: Our reparametrization. We only train A and B.

$$h = W_0x + \Delta Wx = W_0x + BAx$$

$$d \times k = dk$$

$$dr + rk = r(d + k)$$

$$1024 \times 1024 = 1,048,576$$

$$8192 + 8192 = 16384$$

$$W = W_0 + BA$$

# Experiments



LoRA应用于哪些权重矩阵

	# of Trainable Parameters = 18M					
Weight Type	$W_q$	$W_k$	$W_v$	$W_o$	$W_q, W_k$	$W_q, W_v$
Rank $r$	8	8	8	8	4	4
WikiSQL ( $\pm 0.3\%$ )	70.4	70.0	73.0	73.2	71.4	<b>73.7</b>
MultiNLI ( $\pm 0.1\%$ )	91.0	90.8	91.0	<b>91.3</b>	<b>91.3</b>	<b>91.3</b>

	Weight Type	$r = 1$	$r = 2$	$r = 4$	$r = 8$	$r = 64$
WikiSQL ( $\pm 0.3\%$ )	$W_q, W_v$	73.4	73.3	<b>73.7</b>	<b>73.8</b>	73.5
	$W_q$	68.8	69.6	<b>70.5</b>	<b>70.4</b>	70.0
MultiNLI ( $\pm 0.1\%$ )	$W_q, W_v$	91.3	91.4	91.3	<b>91.7</b>	91.4

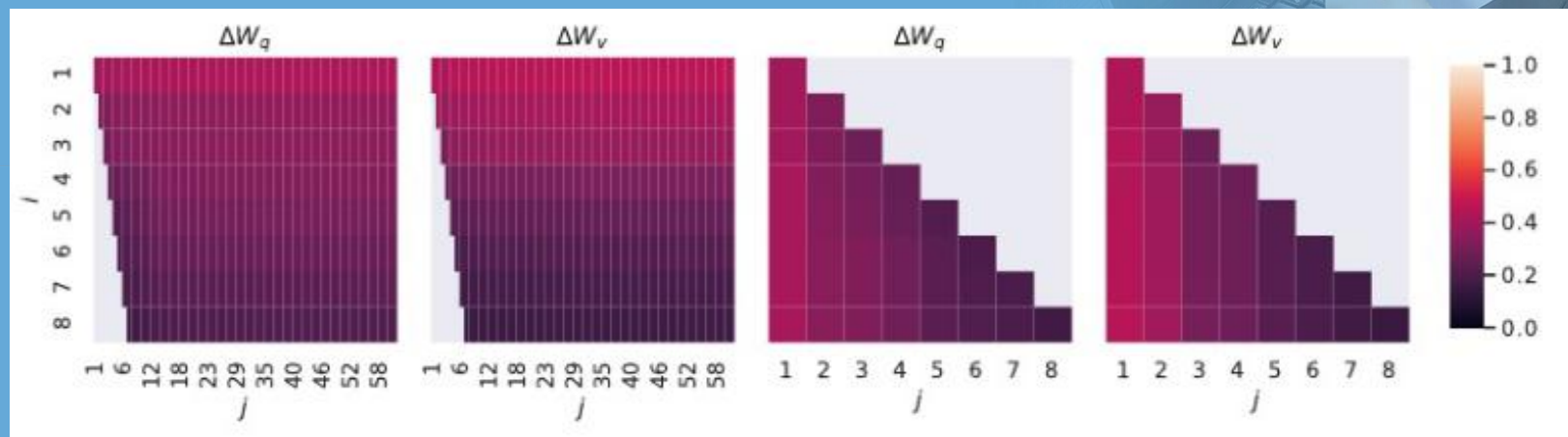
# Experiments



Rank=64 比 Rank=8 多出来的那些方向，并没有带来很多新的核心信息；真正最重要的方向，早在小 rank 的时候就已经学到了

	Weight Type	$r = 1$	$r = 2$	$r = 4$	$r = 8$	$r = 64$
WikiSQL( $\pm 0.3\%$ )	$W_q, W_v$	73.4	73.3	<b>73.7</b>	<b>73.8</b>	73.5
	$W_q$	68.8	69.6	<b>70.5</b>	<b>70.4</b>	70.0
MultiNLI ( $\pm 0.1\%$ )	$W_q, W_v$	91.3	91.4	91.3	<b>91.7</b>	91.4

$$\phi(A_{r=8}, A_{r=64}, i, j) = \frac{\|U_{A_{r=8}}^{i\top} U_{A_{r=64}}^j\|_F^2}{\min(i, j)} \in [0, 1]$$



# Experiments



$\Delta W$ 和 $W$ 之间的关系

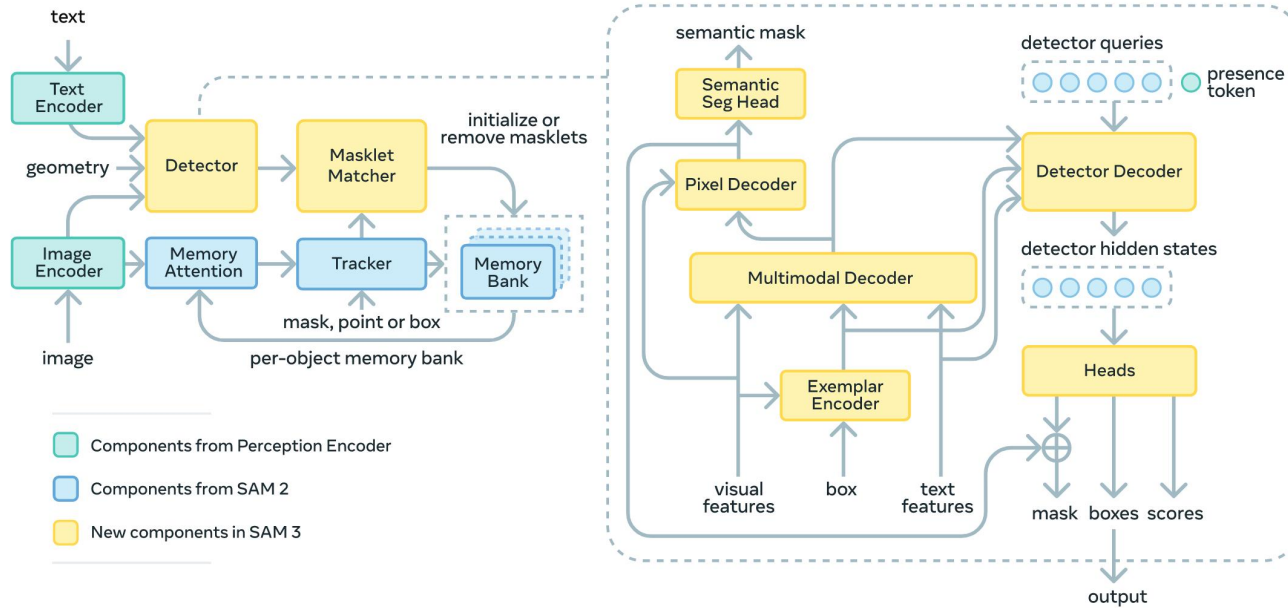
	$r = 4$			$r = 64$		
	$\Delta W_q$	$W_q$	Random	$\Delta W_q$	$W_q$	Random
$\ U^T W_q V^T\ _F =$	0.32	21.67	0.02	1.90	37.71	0.33
$\ W_q\ _F = 61.95$	$\ \Delta W_q\ _F = 6.91$			$\ \Delta W_q\ _F = 3.57$		

我们在GPT-2和GPT-3上对LoRA的下游性能进行了基准测试

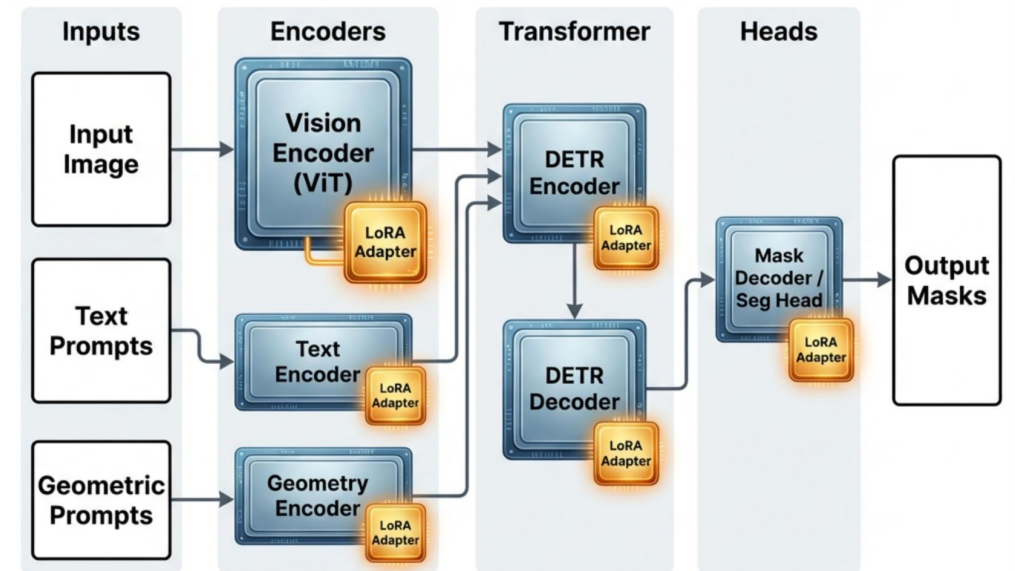
Method	# of Trainable Parameters	WikiSQL	MNLI-m	SAMSum
		Accuracy (%)	Accuracy (%)	R1/R2/RL
GPT-3 175B (Fine-Tune)	175,255.8M	73.0	89.5	52.0/28.0/44.5
GPT-3 175B (Bias Only)	14.2M	71.3	91.0	51.3/27.4/43.5
GPT-3 175B (PrefixEmbed)	3.2M	63.1	88.6	48.3/24.2/40.5
GPT-3 175B (PrefixLayer)	20.2M	70.1	89.5	50.8/27.3/43.5
GPT-3 175B (LoRA)	4.7M	73.4	91.3	52.1/28.3/44.3
GPT-3 175B (LoRA)	37.7M	<b>73.8</b>	<b>91.7</b>	<b>53.2/29.2/45.0</b>

Method	# of Trainable Parameters	BLEU	NIST	E2E MET	ROUGE-L	CIDEr
GPT-2 M (Fine-Tune)	354.92M	68.2	8.62	46.2	71.0	2.47
GPT-2 M (Adapter)	11.48M	68.9	8.71	46.1	71.3	2.47
GPT-2 M (FT-Top2)	25.19M	68.1	8.59	46.0	70.8	2.41
GPT-2 M (Prefix)	0.35M	69.7	8.81	46.1	71.4	2.49
GPT-2 M (LoRA)	0.35M	<b>70.4</b>	<b>8.85</b>	<b>46.8</b>	<b>71.8</b>	<b>2.53</b>
GPT-2 L (Fine-Tune)	774.03M	68.5	8.78	46.0	69.9	2.45
GPT-2 L (Prefix)	0.77M	70.3	8.85	46.2	71.7	<b>2.47</b>
GPT-2 L (LoRA)	0.77M	<b>70.4</b>	<b>8.89</b>	<b>46.8</b>	<b>72.0</b>	<b>2.47</b>

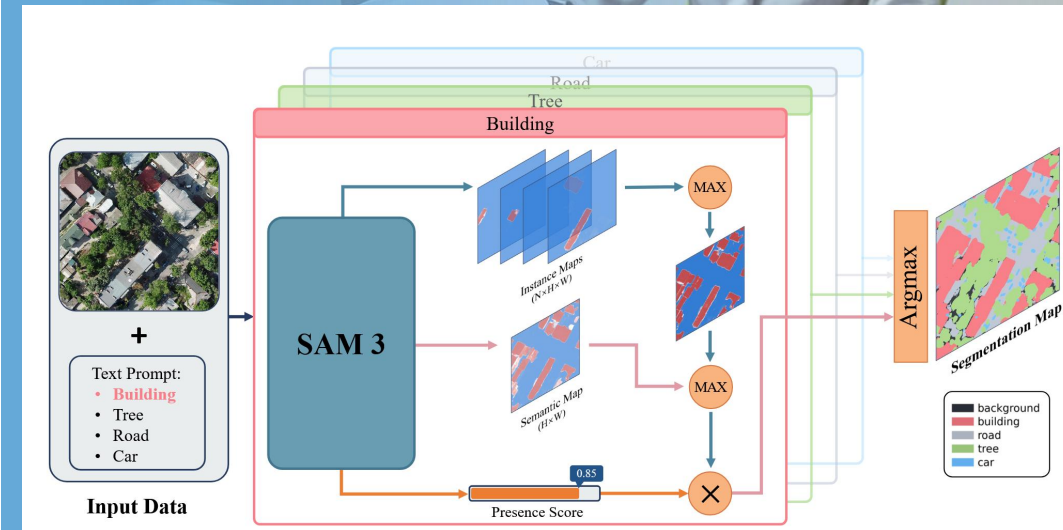
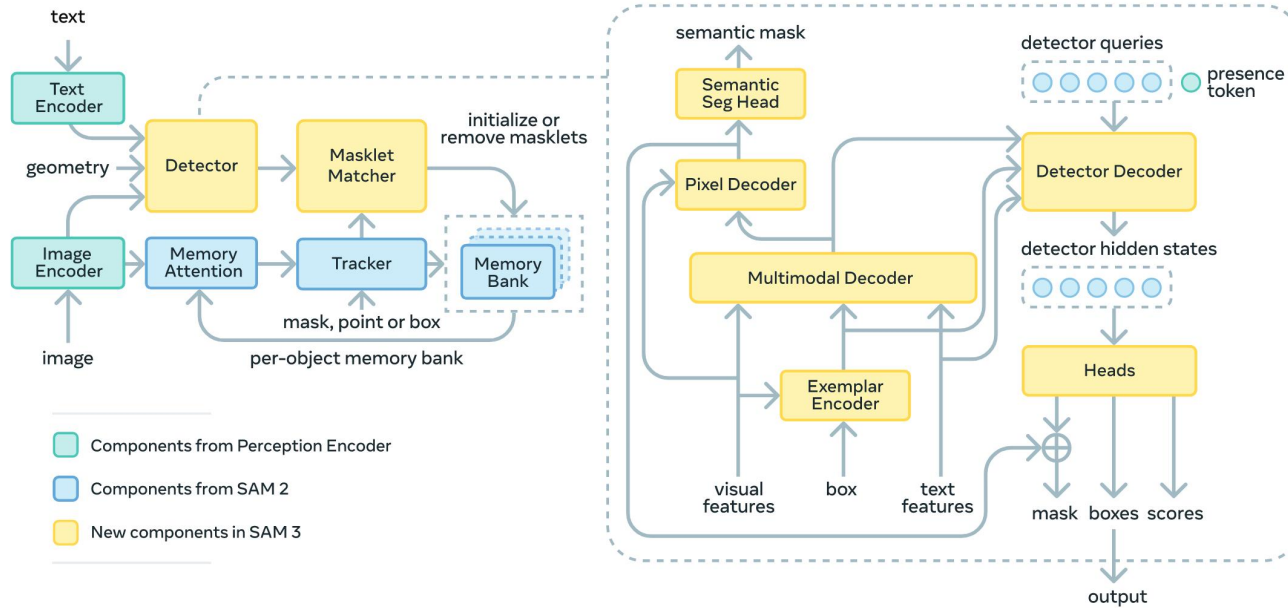
# Method



## SAM3 Model Architecture with Full LoRA Adaptation



# Method



aAcc: 49.6400 mIoU: 35.3900 mAcc: 53.1700

Class	IoU	Acc
background	34.62	60.85
building	50.12	71.46
road	51.0	67.71
water	36.09	38.56
barren	21.55	54.37
forest	28.4	47.94
agricultural	25.92	31.28

aAcc: 68.1000 mIoU: 50.7900 mAcc: 63.0400

Class	IoU	Acc
background	50.88	83.43
building	65.11	82.25
road	56.74	64.28
water	64.51	74.0
barren	29.55	38.59
forest	38.31	44.81
agricultural	50.4	53.9

aAcc: 66.6400 mIoU: 50.0100 mAcc: 65.8100

Class	IoU	Acc
background	48.92	74.48
building	61.99	83.06
road	56.03	62.06
water	63.19	74.91
barren	27.42	56.65
forest	42.51	55.09
agricultural	50.0	54.45

aAcc: 63.8000 mIoU: 47.3900 mAcc: 62.0100

Class	IoU	Acc
background	45.5	69.41
building	63.81	78.6
road	53.88	70.5
water	51.43	54.71
barren	35.77	53.92
forest	33.86	44.95
agricultural	47.48	61.99



*Thanks*

