ReFT: Representation Finetuning for Language Models

Contents Covered Here

- **What is PEFT? and Why PEFT is needed in the first place**
- Types of PEFT methods
  - Adding extra layer sequentially (general finetuning)
  - Finetuning with Minimal Layers
    - LoRA
    - DoRA
- Quantization methods
- ReFT methodology
- Conclusion and Future enhancements from here (personal note)

What is PEFT? and Why PEFT is needed in the first place

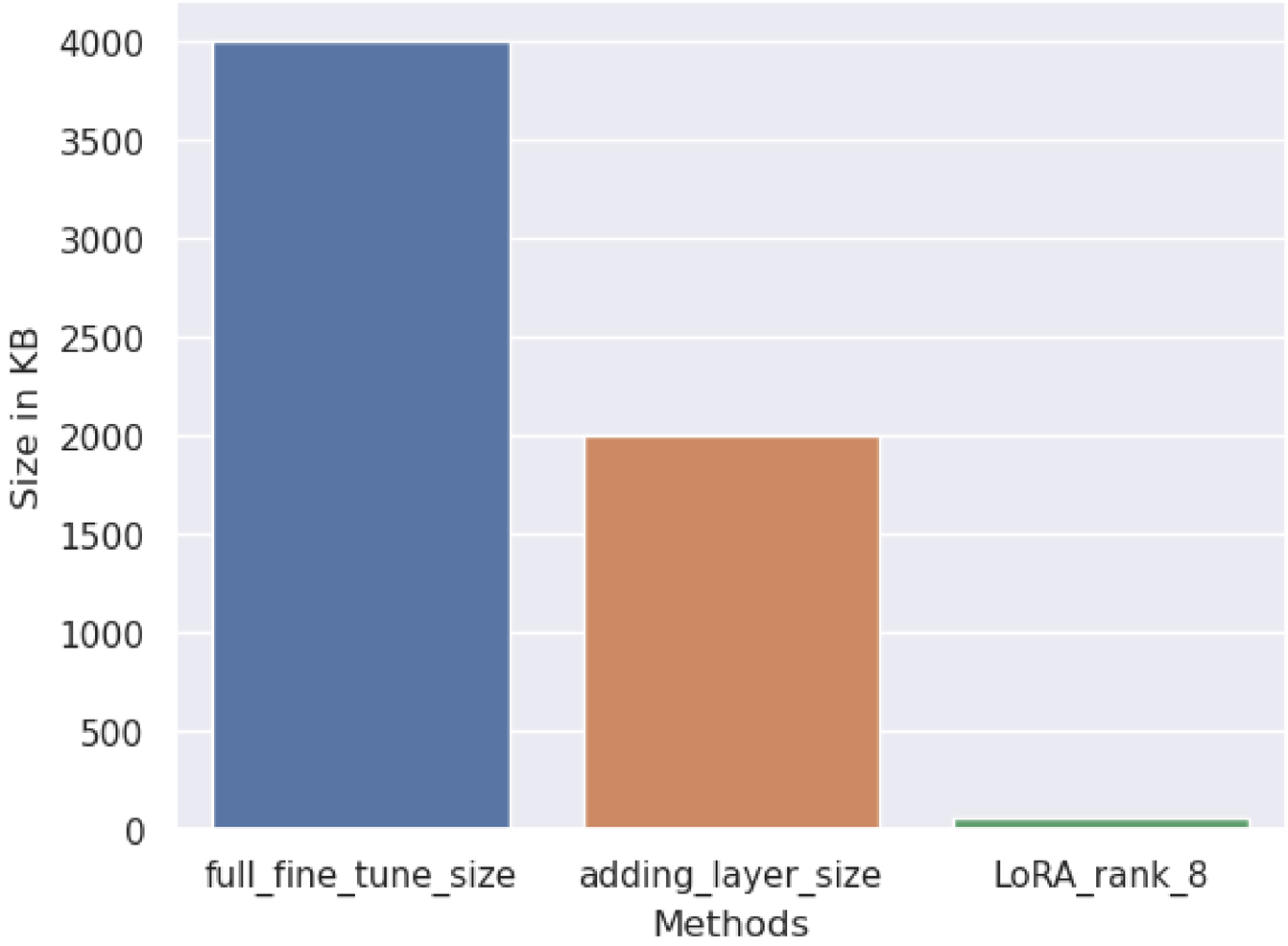
- PEFT stands for Parametric Efficient Fine Tuning which involves in training a Parameter heavy models (such as LLM) with minimal parameters as possible without loosing the originality of the model.

- As LLMs are getting heavier and heavier it is nearly impossible to finetune the entire model like conventional DL models (considering adequate amount of data is present) on one GPU

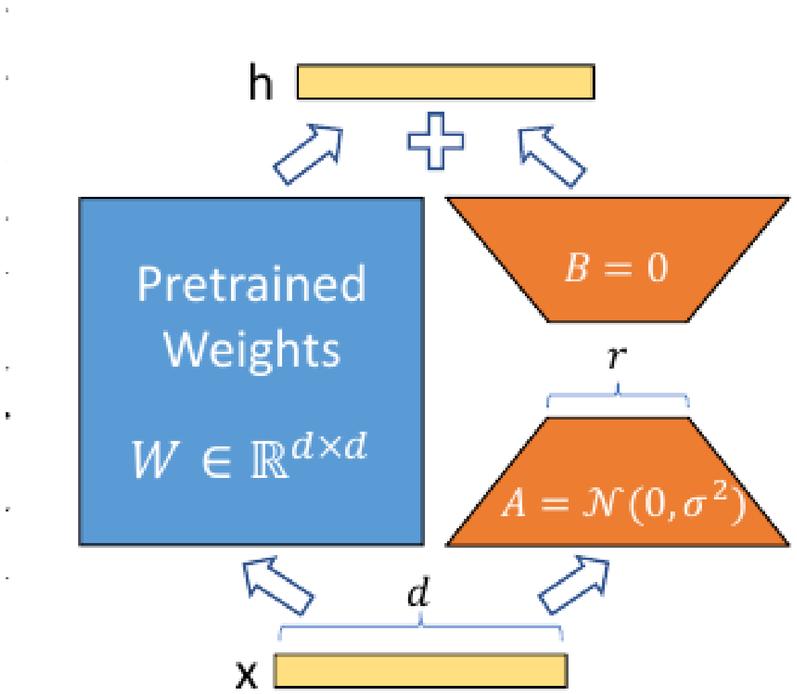
- So to fine tune the model on one bigger GPU or on multiple smaller GPUs we need PEFT

Practical example for this

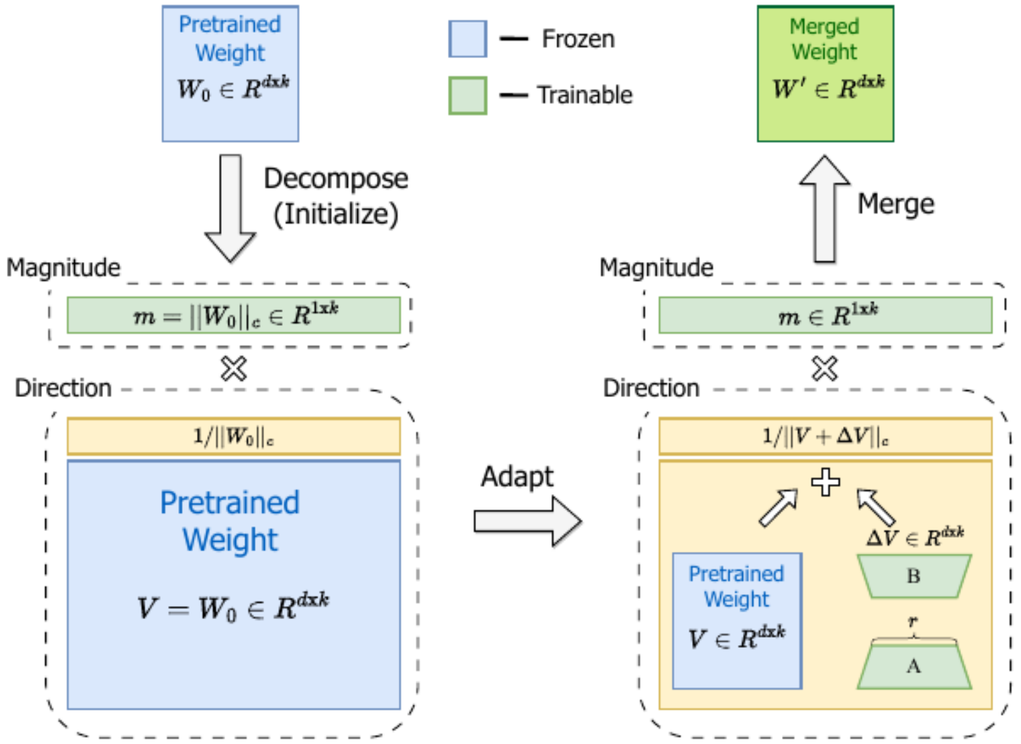
Consider a Matrix of size 1000 x 1024 x 1024 with fp32 precision



LoRA


h
Pretrained Weights
$W \in \mathbb{R}^{d \times d}$
$B = 0$
r
$A = \mathcal{N}(0, \sigma^2)$
d
x

DoRA


Pretrained Weight
$W_0 \in R^{dxk}$
Frozen
Trainable
Merged Weight
$W' \in R^{dxk}$
Decompose (Initialize)
Merge
Magnitude
$m = ||W_0||_c \in R^{1xk}$
Magnitude
$m \in R^{1xk}$
Direction
$1/||W_0||_c$
Pretrained Weight
$V = W_0 \in R^{dxk}$
Adapt
Direction
$1/||V + \Delta V||_c$
$\Delta V \in R^{dxk}$
B
r
A
Pretrained Weight
$V \in R^{dxk}$

DoRA - Inference

- In this you scale the Weight matrix into Unit vector and try making the magnitude of the matrix m as a learnable parameter

Advantages

- Can change its slope towards the negative directions and algin with representations learnt from fully finetuning the model

$$W = m\frac{V}{||V||_c} = ||W||_c\frac{W}{||W||_c}$$

DoRA Inference

- We make the Magnitude learnable and $V^{'} = V + \triangle V$

$$W' = \underline{m} \frac{V + \Delta V}{||V + \Delta V||_c} = \underline{m} \frac{W_0 + \underline{BA}}{||W_0 + \underline{BA}||_c}$$

$$\nabla_{V'} \mathcal{L} = \frac{m}{||V'||_c} \left(I - \frac{V' V'^{\mathbf{T}}}{||V'||_c^2} \right) \nabla_{W'} \mathcal{L}$$

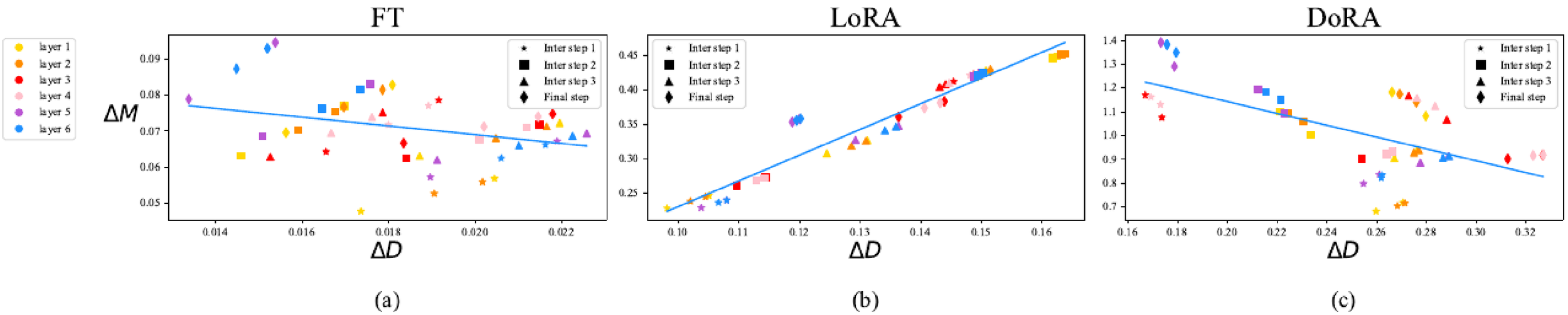
$$\nabla_m \mathcal{L} = \frac{\nabla_{W'} \mathcal{L} \cdot V'}{||V'||_c}$$

$$d\|\mathbf{x}\| = \frac{\mathbf{x}^T d\mathbf{x}}{\|\mathbf{x}\|}$$

Why DoRA With Stop Gradients

$$\nabla_{V'}\mathcal{L} = \frac{m}{C}\nabla_{W'}\mathcal{L} \text{ where } C = ||V'||_c$$

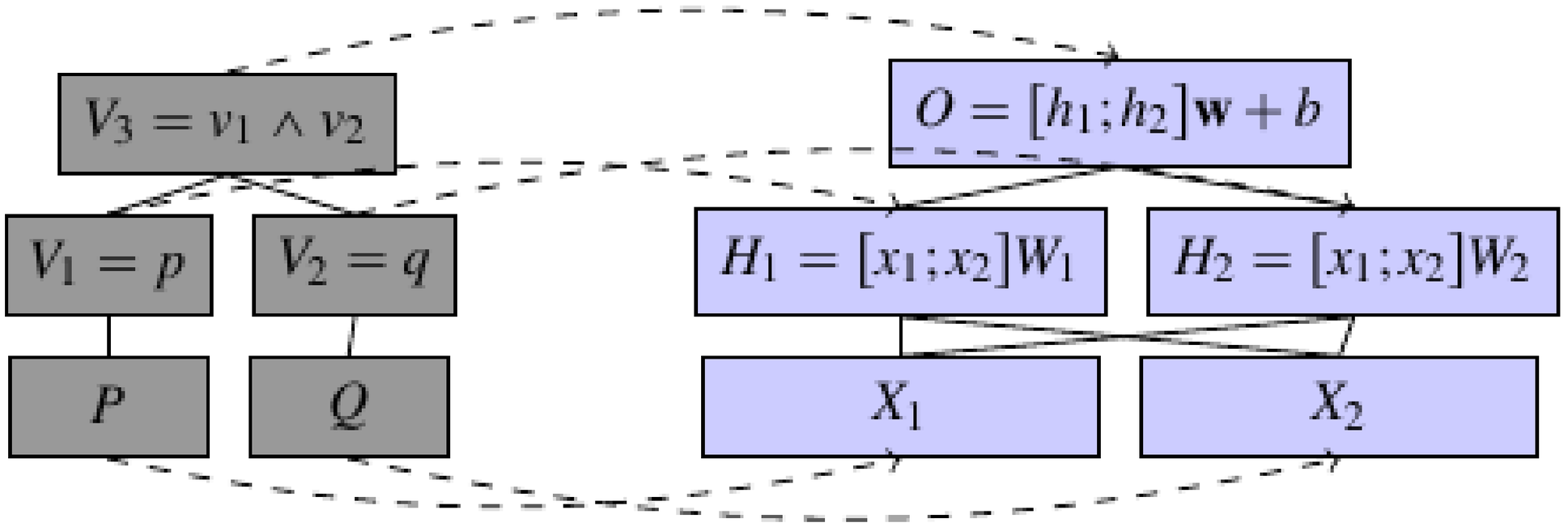
Why DoRA Works ? and if so why it is better than LoRA



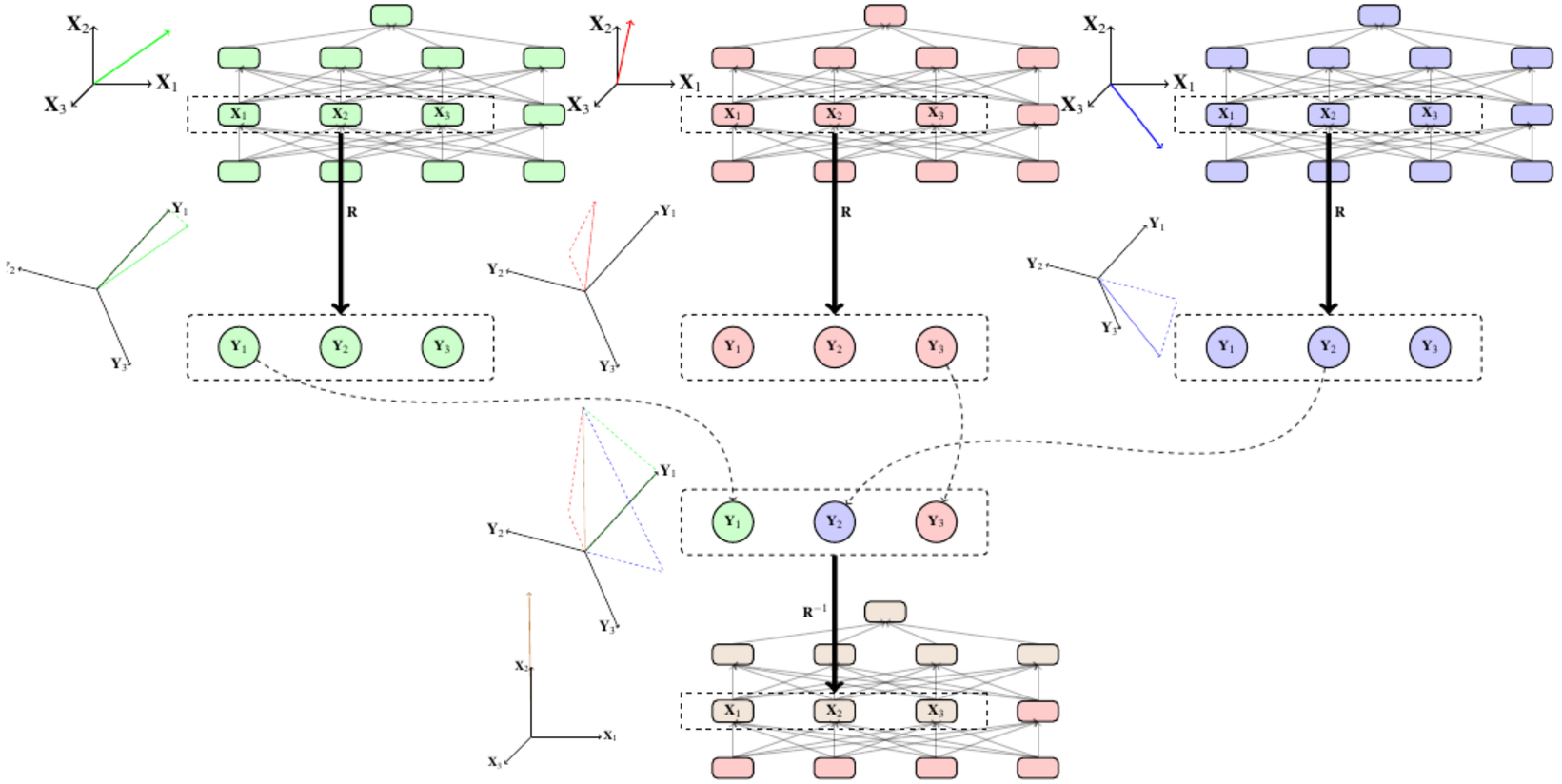
Prerequestite to ReFT

- DII (Distributed interchange intervention)
- DAS (Distributed Alignment Search)

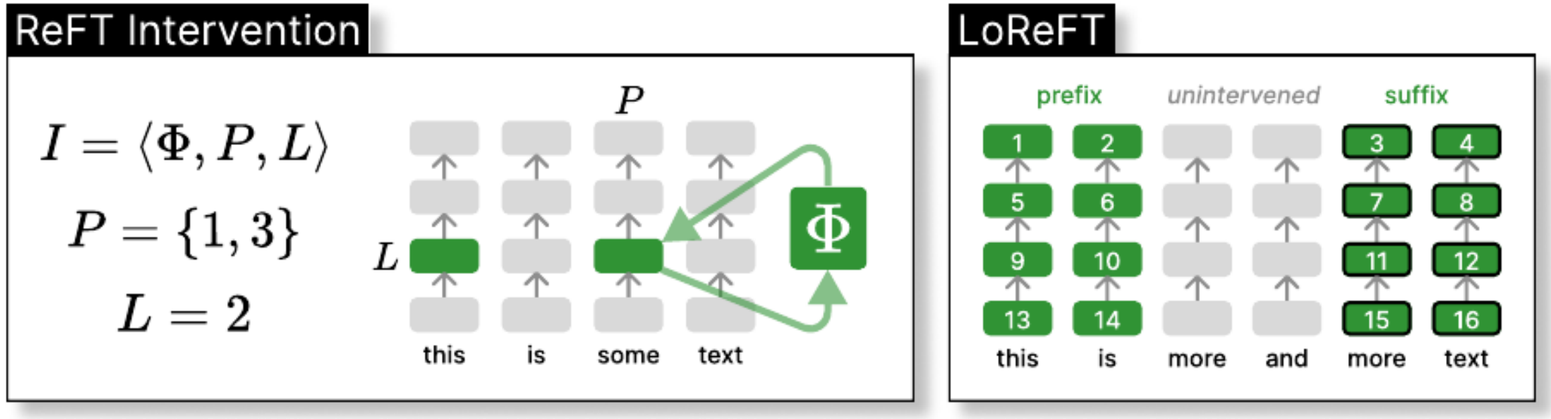
DII and DAS


$V_3 = v_1 \wedge v_2$
$V_1 = p$
$V_2 = q$
P
Q
$O = [h_1; h_2]\mathbf{w} + b$
$H_1 = [x_1; x_2]W_1$
$H_2 = [x_1; x_2]W_2$
X_1
X_2

DII and DAS



ReFT


ReFT Intervention
$I = \langle \Phi, P, L \rangle$
$P = \{1, 3\}$
$L = 2$
P
L
Φ
this
is
some
text
LoReFT
prefix
unintervened
suffix
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
this
is
more
and
more
text