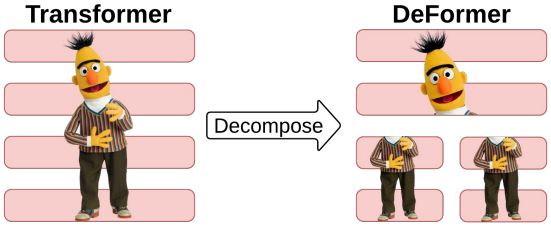
DeFormer: Decomposing Pre-trained Transformers for Faster Question Answering



Qingqing Cao

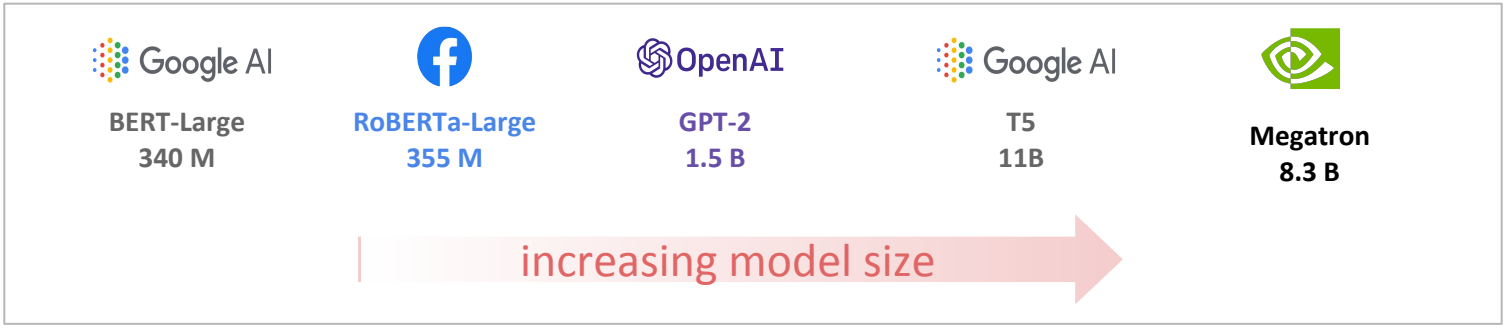
Harsh Trivedi

Aruna Balasubramanian

Niranjan Balasubramanian

Stony Brook University

Resource Challenges in Large Transformers



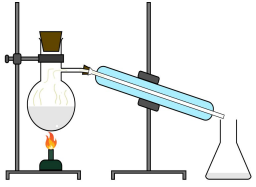
High runtime latency

10 s to answer a question!

High runtime memory

> half GB to run **one** question

How to Reduce Runtime Latency and Memory?



Distillation

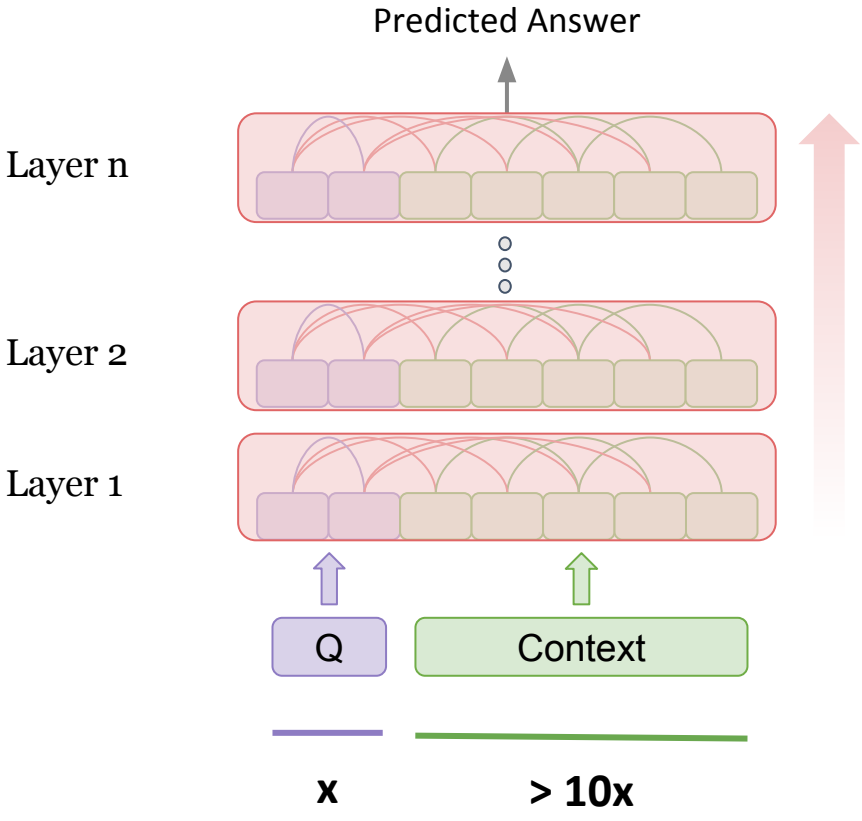
expensive pre-training 😟

DistillBERT, TinyBERT ...

orthogonal approach to run faster?

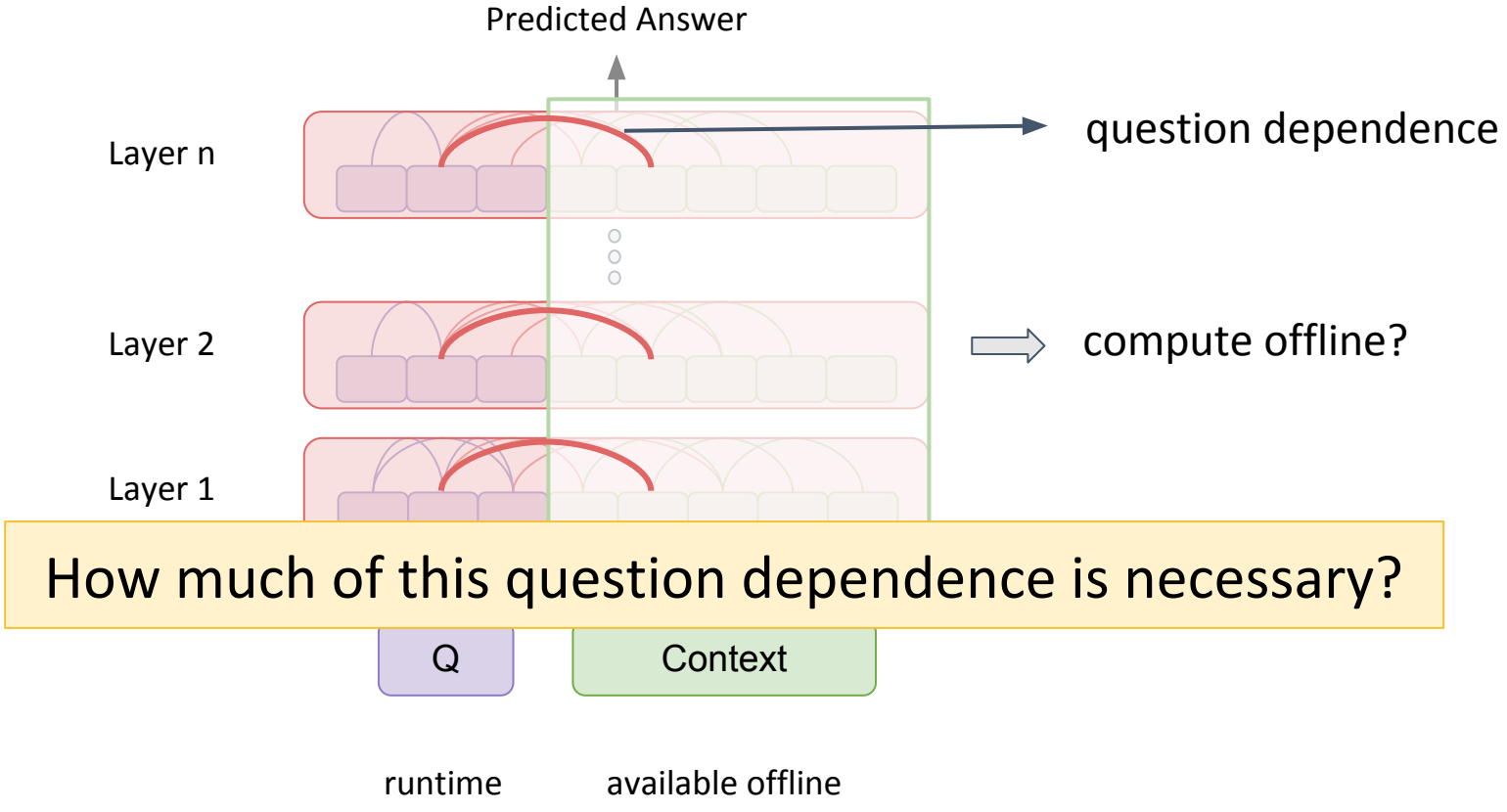
no pre-training 😀

Why are Transformers Slow for QA?

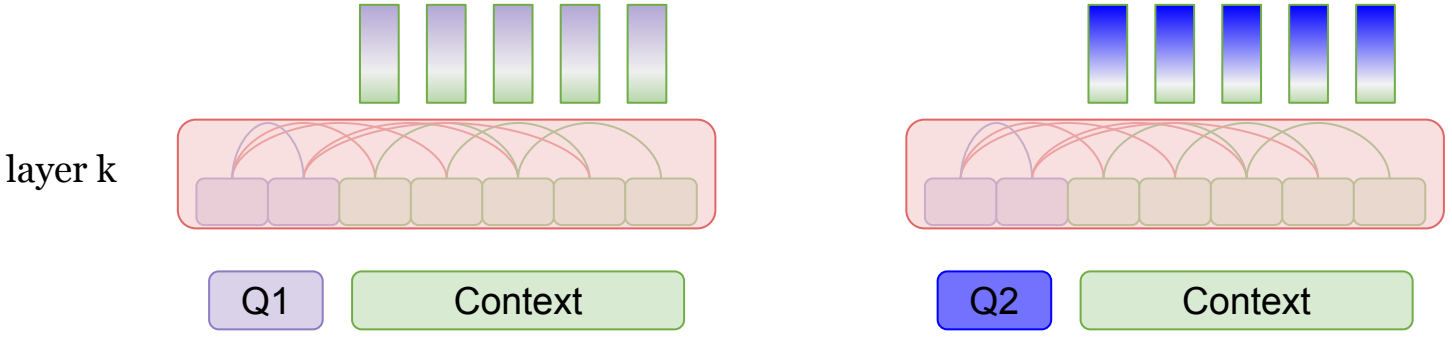


- Context is much longer than the question
- Context processing is the bottleneck in all layers

How Can We Reduce the Context Bottleneck?

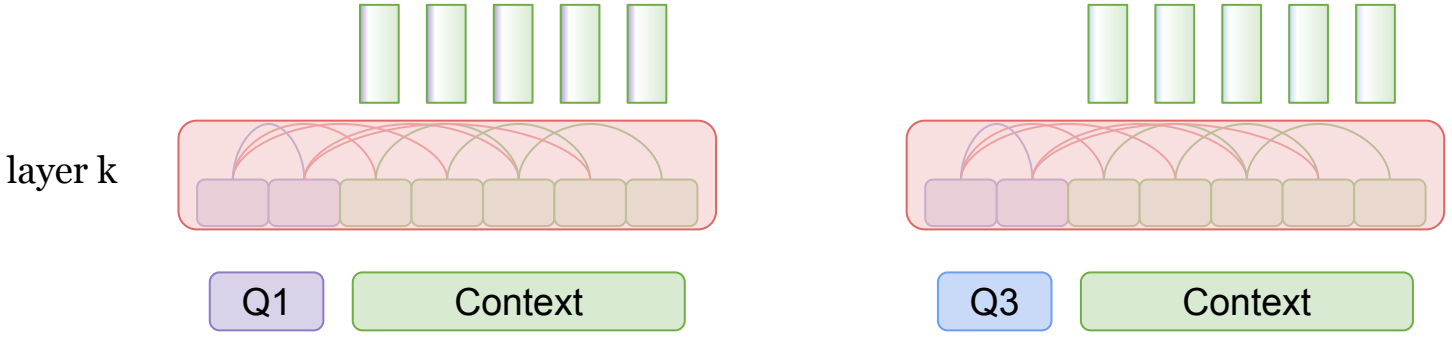


How critical is question dependence?



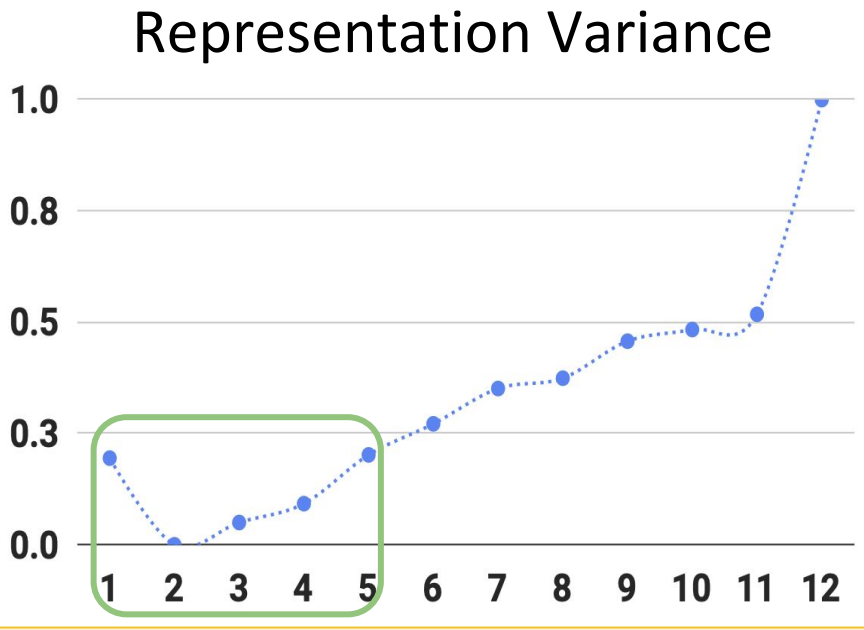
High question dependence

How critical is question dependence?



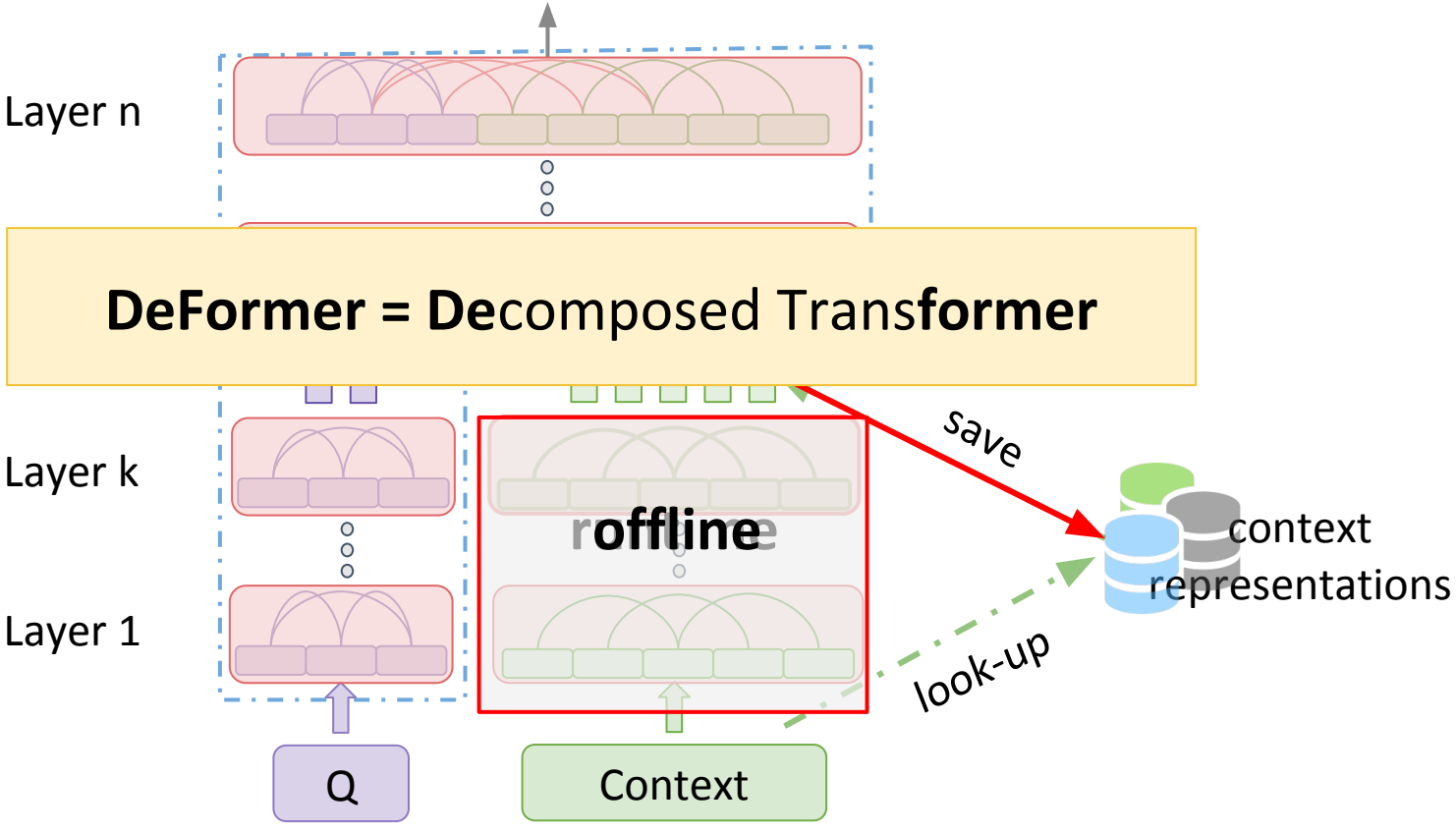
Less question dependence

Question Dependence in Different Layers

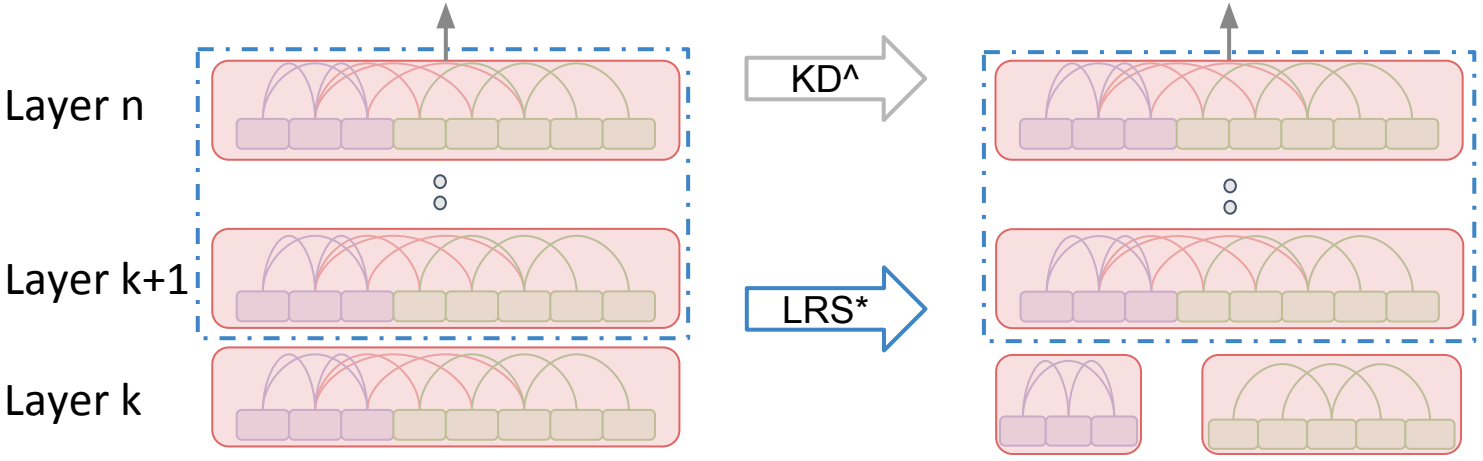


Question dependence is less in the lower layers!

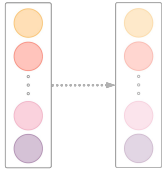
Decomposing the Dependence in Lower Layers



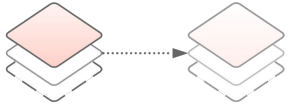
Fine-tuning DeFormer with Auxiliary Supervision



^knowledge distillation



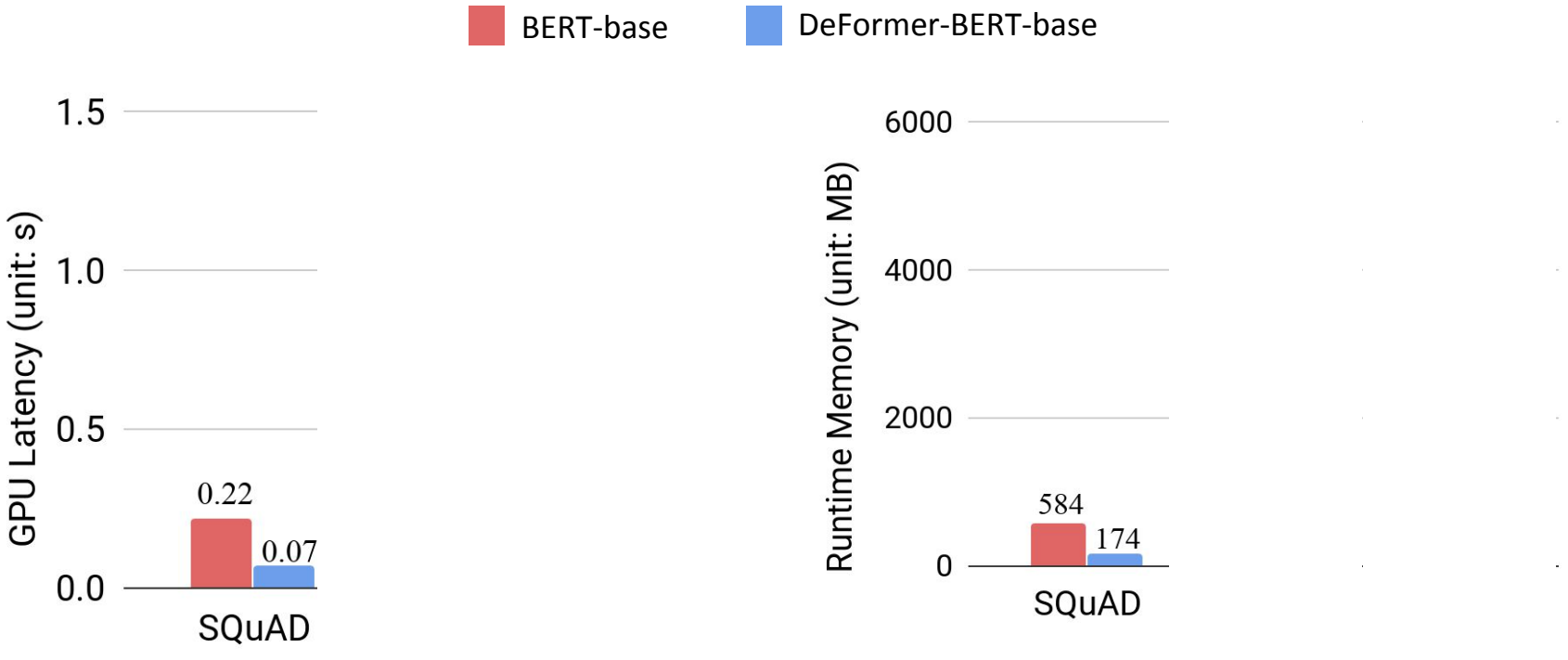
*layerwise representation similarity



Evaluation

- How effective is DeFormer in latency, memory and accuracy?
- What is the impact of auxiliary supervision?
- What is the efficiency tradeoff at different layers?

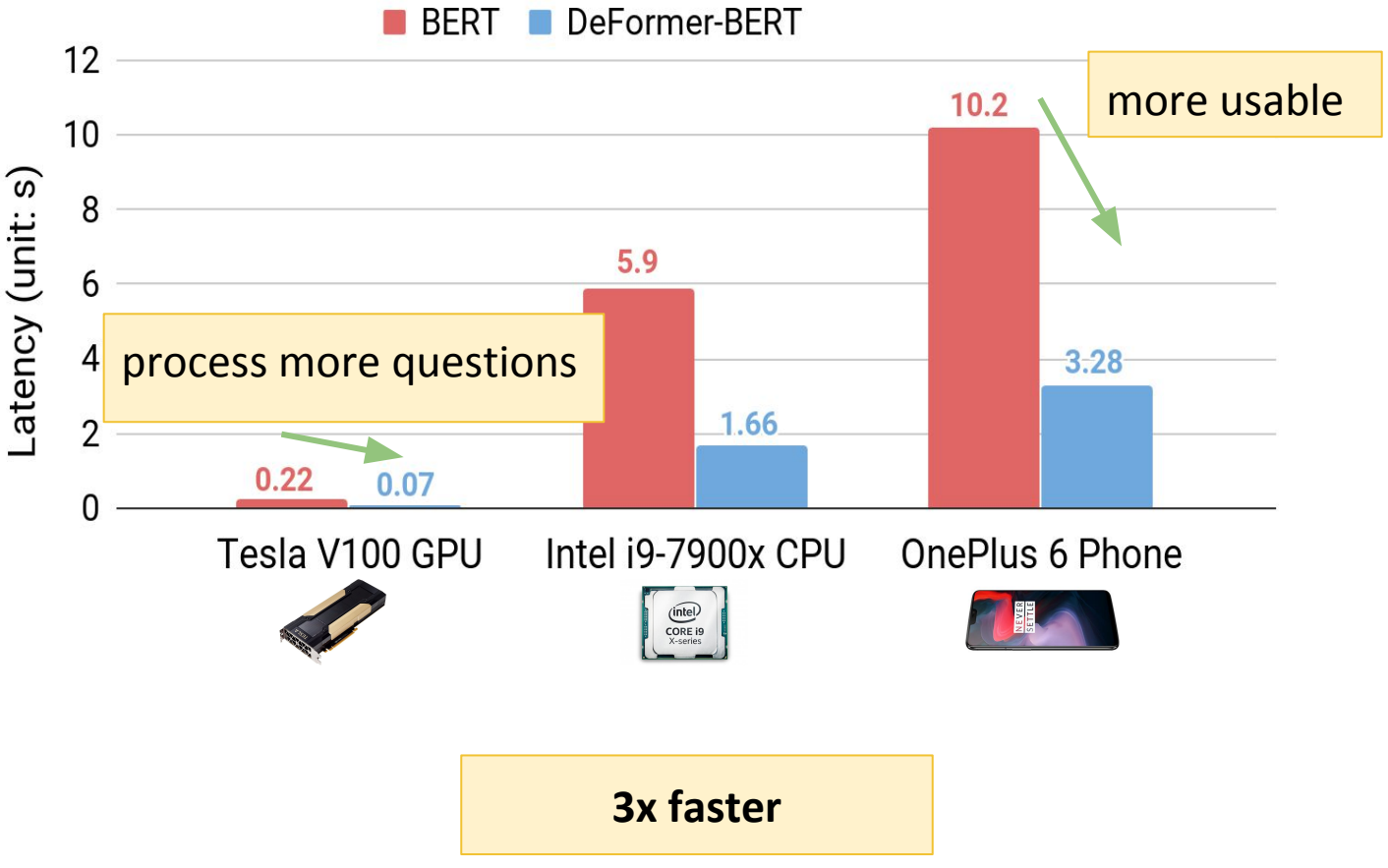
DeFormer is Faster and Memory Efficient



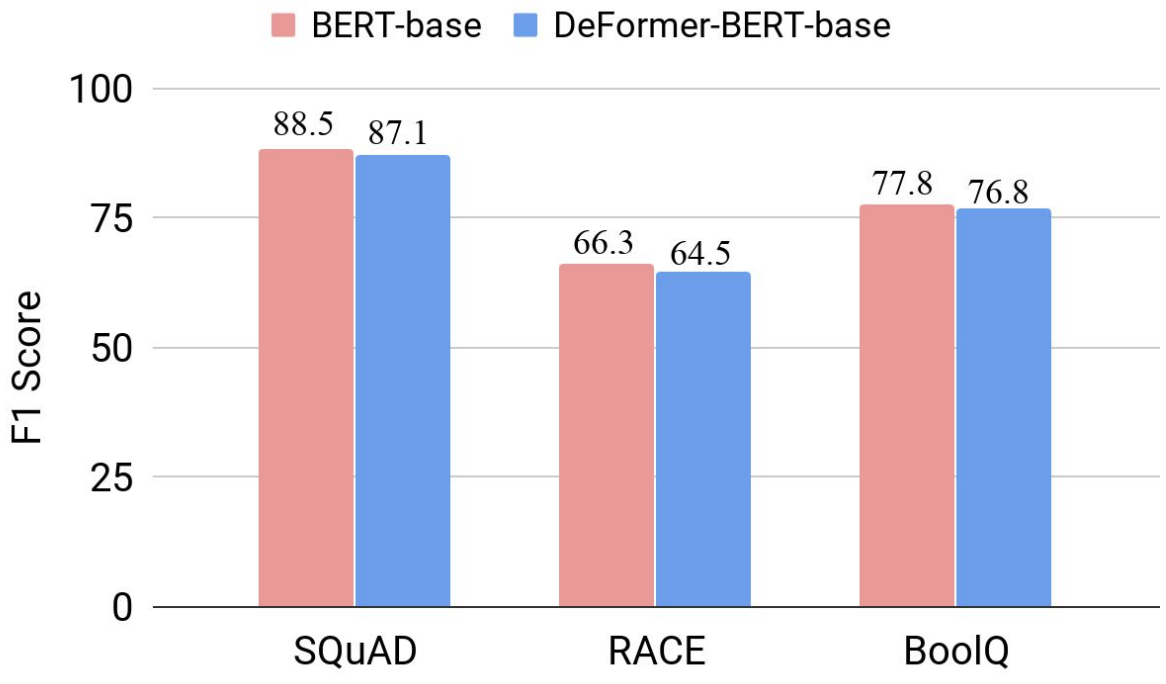
3x reduction in runtime latency

70% reduction in runtime memory

DeFormer Gains Speedup on Diverse Devices

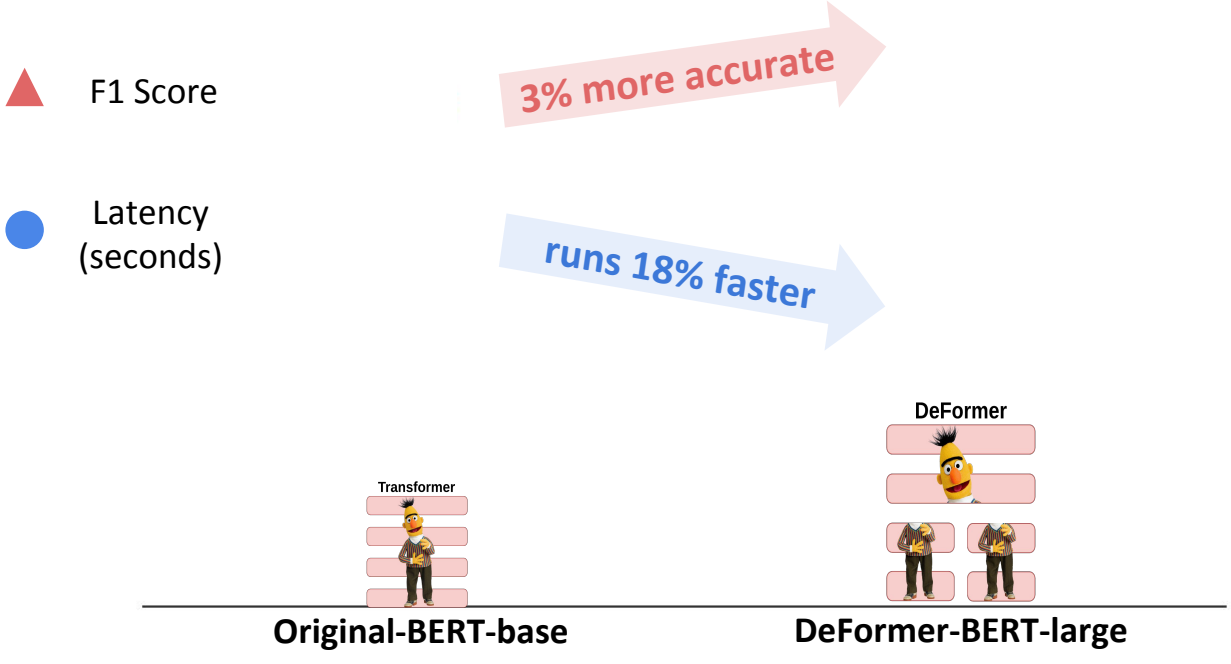


DeFormer (Mostly) Retains Original Performance



DeFormer retains **98%** of the original model's performance

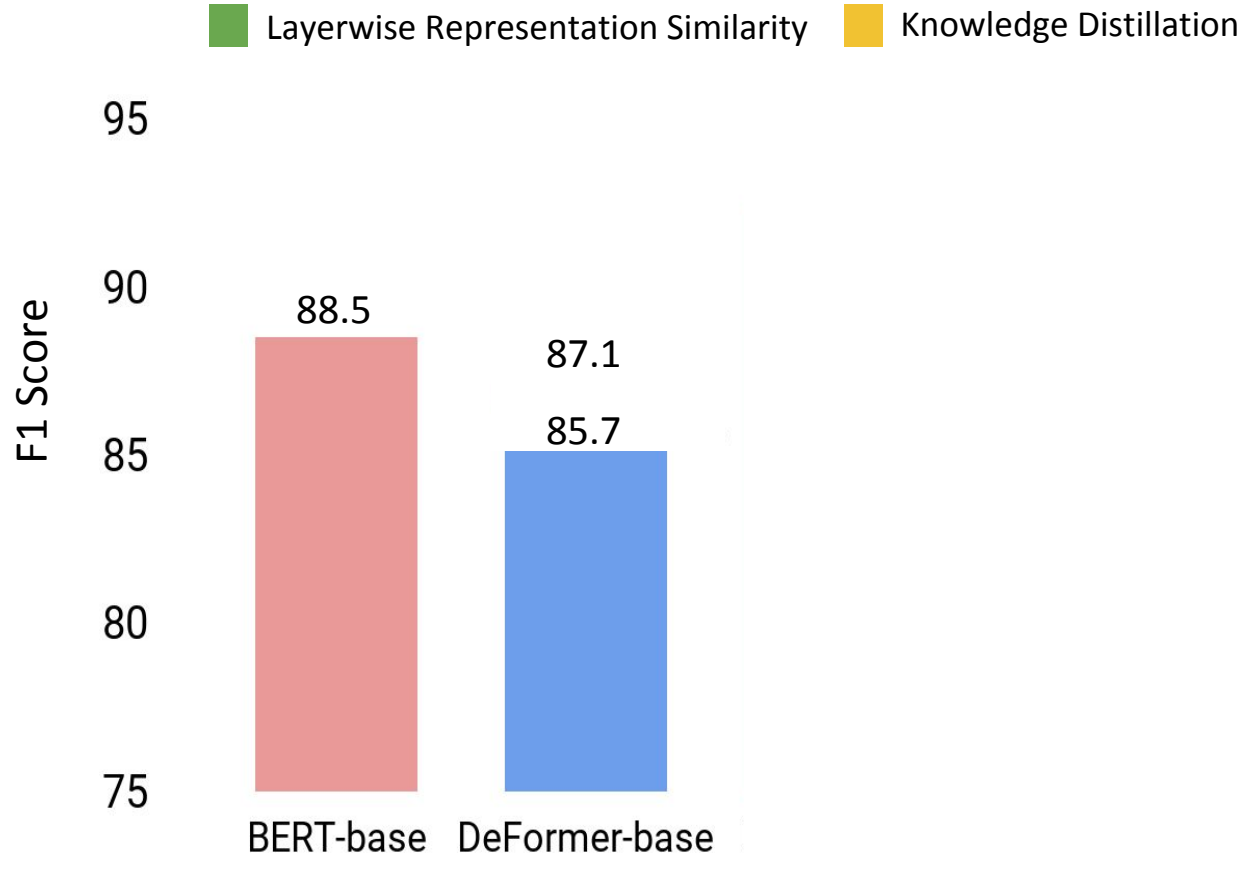
A Large DeFormer is Faster and More Accurate than a Smaller Transformer



Evaluation

- How effective is DeFormer in latency, memory and accuracy?
- What is the impact of auxiliary supervision?
- What is the efficiency tradeoff at different layers?

LRS and KD both Provide Benefits in DeFormer



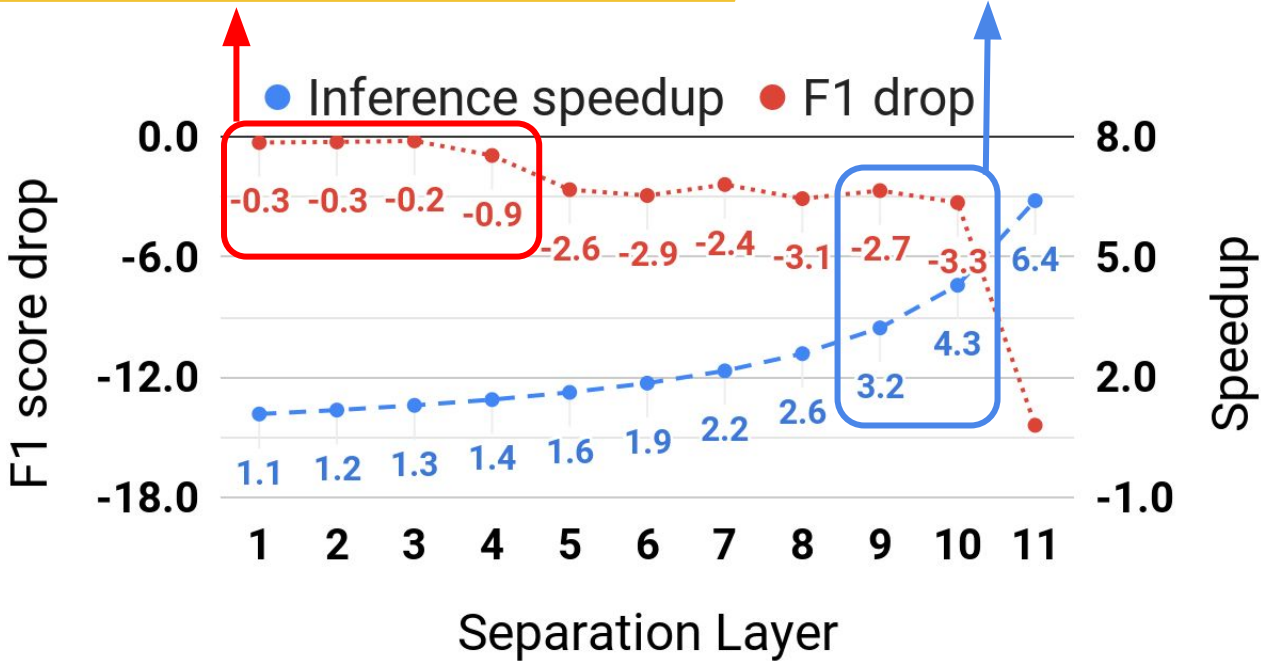
Evaluation

- How effective is DeFormer in latency, memory and accuracy?
- What is the impact of auxiliary supervision?
- What is the efficiency tradeoff at different layers?

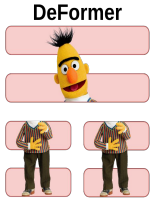
Which Layer to Decompose?

less dependence, less performance impact

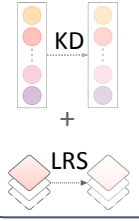
good tradeoff points



Takeaways



DeFormer uses a simple decomposition technique that significantly speeds up inference, reduces memory while retaining most of the model's accuracy



DeFormer leverages distillation along with auxiliary supervision to reduce performance gap to the original Transformer

DeFormer remains largely the same as the Transformer, allowing reusing original model weights without repeating pre-training