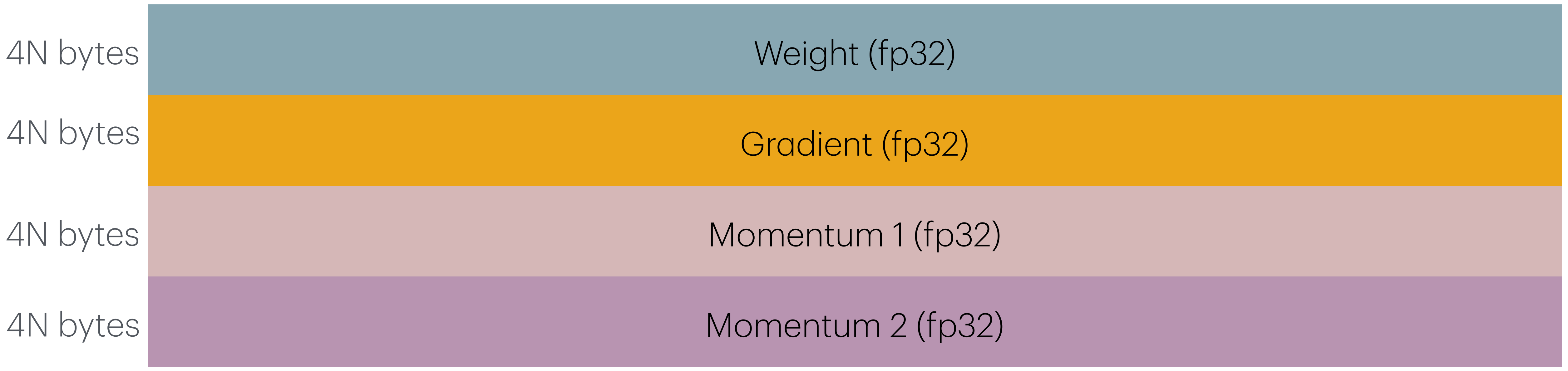
Quantized Low-rank adapters

Philipp Krähenbühl, UT Austin

Training large models

Memory requirements

- Without optimization:
  - Model parameters: N
  - Weights: N floats
  - Gradients: N floats
  - Momentum: N floats
  - 2nd momentum (ADAM): N floats
- 16N bytes without counting activations



Inference in large models

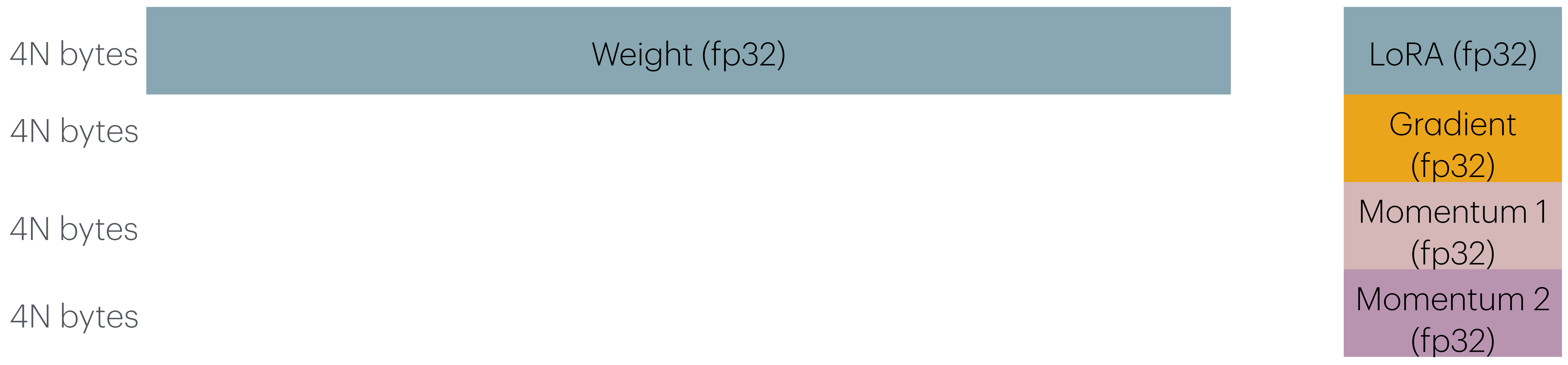
Memory requirements

- Quantization:
  - Model parameters: N
  - Weights: N int4/int8
- $\frac{1}{2}$N bytes without counting activations

Training LoRA models

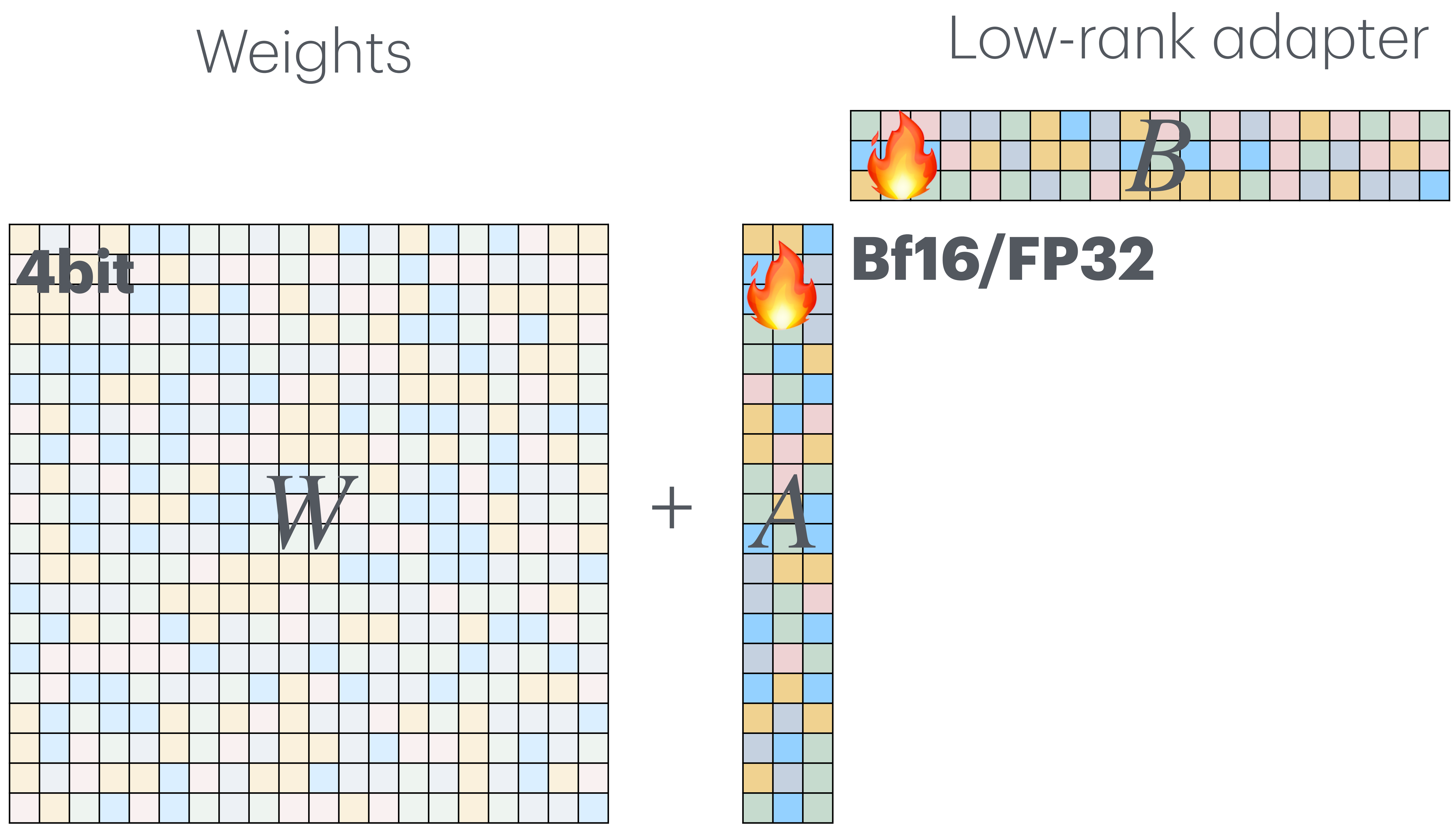
Memory requirements

- LoRA
  - Model parameters: N, LoRA param M
  - Weights: N+M floats
  - Gradients: M floats
  - Momentum: M floats
  - 2nd momentum (ADAM): M floats
- 4N+16M bytes without activations
- M often ~1-5% of N



QLoRA

- Train a LoRA adapter on a quantized model
- Quantize the weights
  - Requires a pre-trained model
- Learn an adapter in high precision

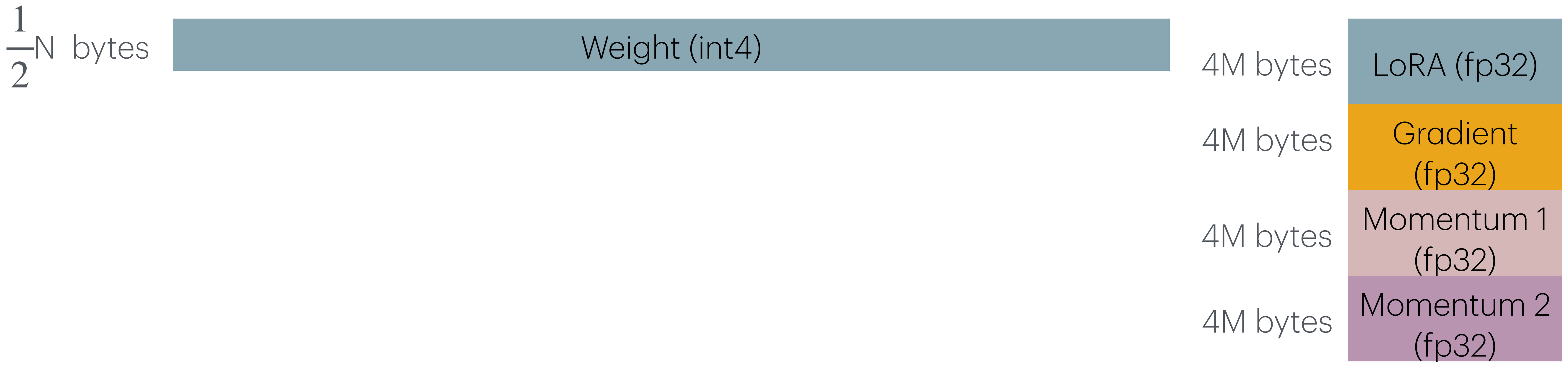


[1] Tim Dettmers, et al. QLoRA: Efficient Finetuning of Quantized LLMs. 2023

Training QLoRA models

Memory requirements

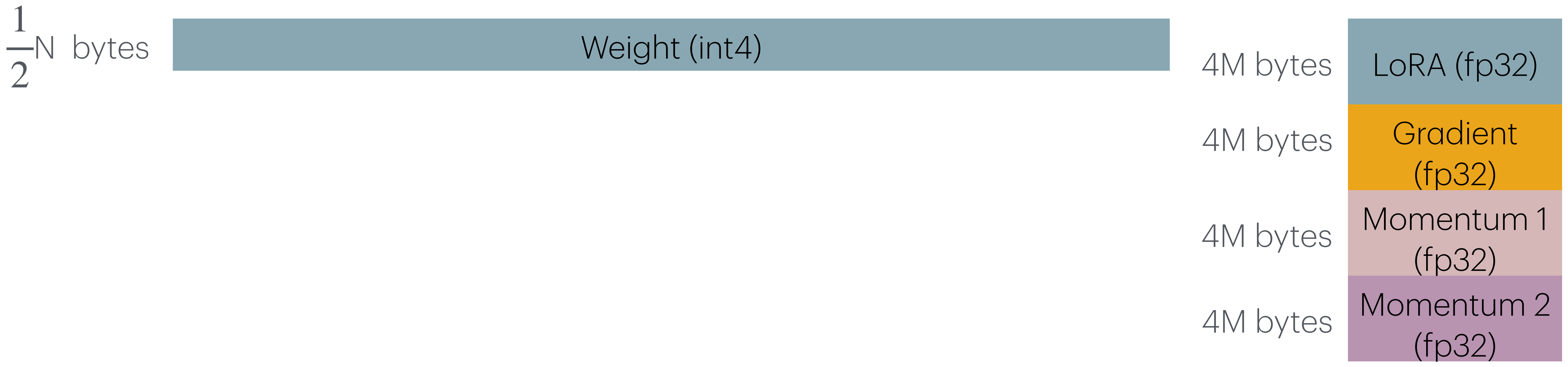
- QLoRA
    - Model parameters: N, LoRA param M
    - Weights: N int4, M floats
    - Gradients: M floats
    - Momentum: M floats
    - 2nd momentum (ADAM): M floats
- $\frac{1}{2}$N+16M bytes without activations
- M often ~1-5% of N



QLoRA

Tradeoffs

- Advantage
  - Extremely low optimizer memory
- Disadvantages
  - Fine-tuning only (no pre-training)
  - Task dependent
    - May require large rank R



[1] Tim Dettmers, et al. QLoRA: Efficient Finetuning of Quantized LLMs. 2023

References

- [1] Tim Dettmers, et al. QLoRA: Efficient Finetuning of Quantized LLMs. 2023. (link)