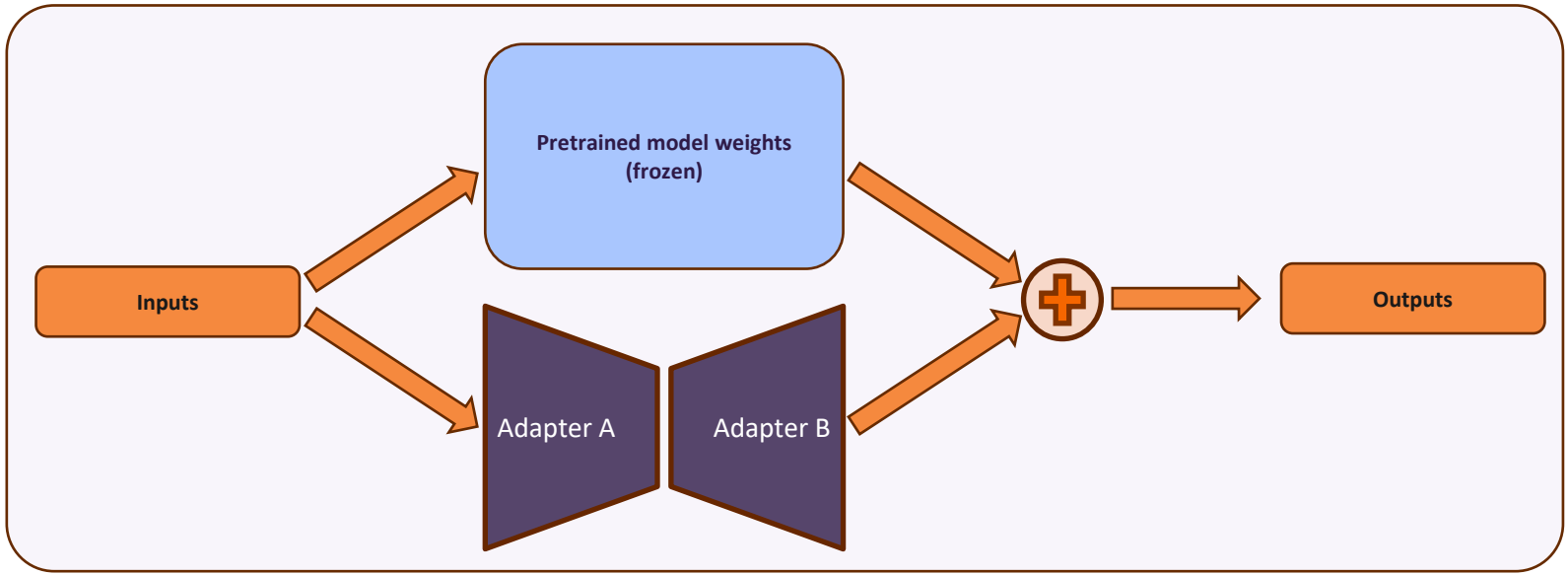
Low-rank adaptation (LoRA) fine-tune

Adds small trainable adapter layers, and trains *only* these new layers.

Reduces memory and compute significantly, and reduces catastrophic forgetting.

Usually best.

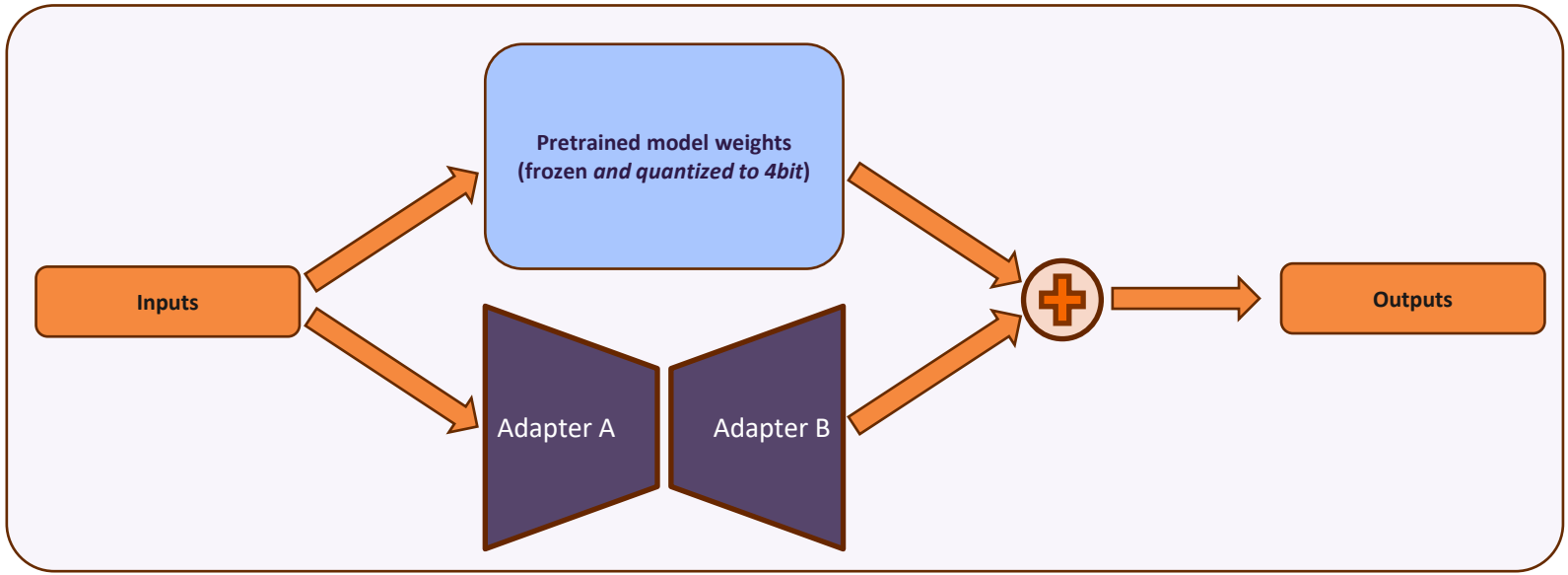


QLoRA fine-tune

Just like LoRA, but also quantizes the model to 4-bit precision, further reducing memory requirements.

Allows for fine-tuning models otherwise too large to fit in available memory.

Best when the available hardware isn't large enough to just do LoRA.

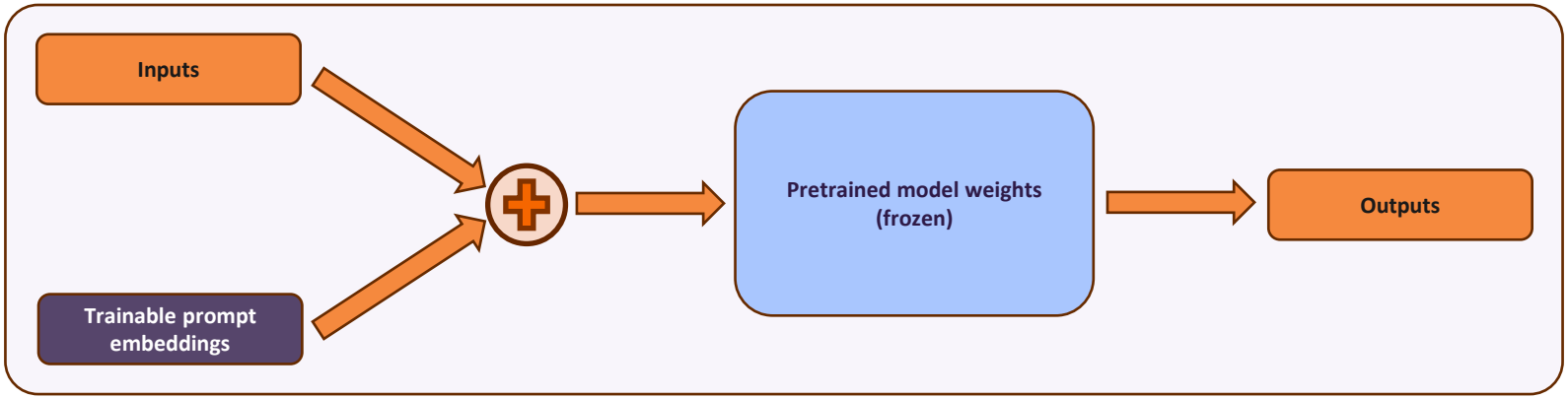


Prompt tuning

Instead of modifying the model weights, learns small trainable prompt embeddings.

Less powerful than LoRA (sometimes a good thing!) and very efficient.

Best for: task adaptation without modifying the model.



Fine-tuning methods comparison

Method	Trainable parameters	Computational cost	Memory usage	Retains general knowledge?	Best use cases
Full fine-tuning	All model parameters	High	Very high	Risk of catastrophic forgetting	Task-specific, highly customized models
LoRA	Small adapter layers	Low	Moderate	Yes	Domain adaptation, style control
QLoRA	Small adapter layers	Very low	Very low	Yes	Fine-tuning very large LLMs (relative to hardware)
Prompt tuning	Very small prompt embeddings	Low	Low	Yes	Task adaptation with minimal change to model