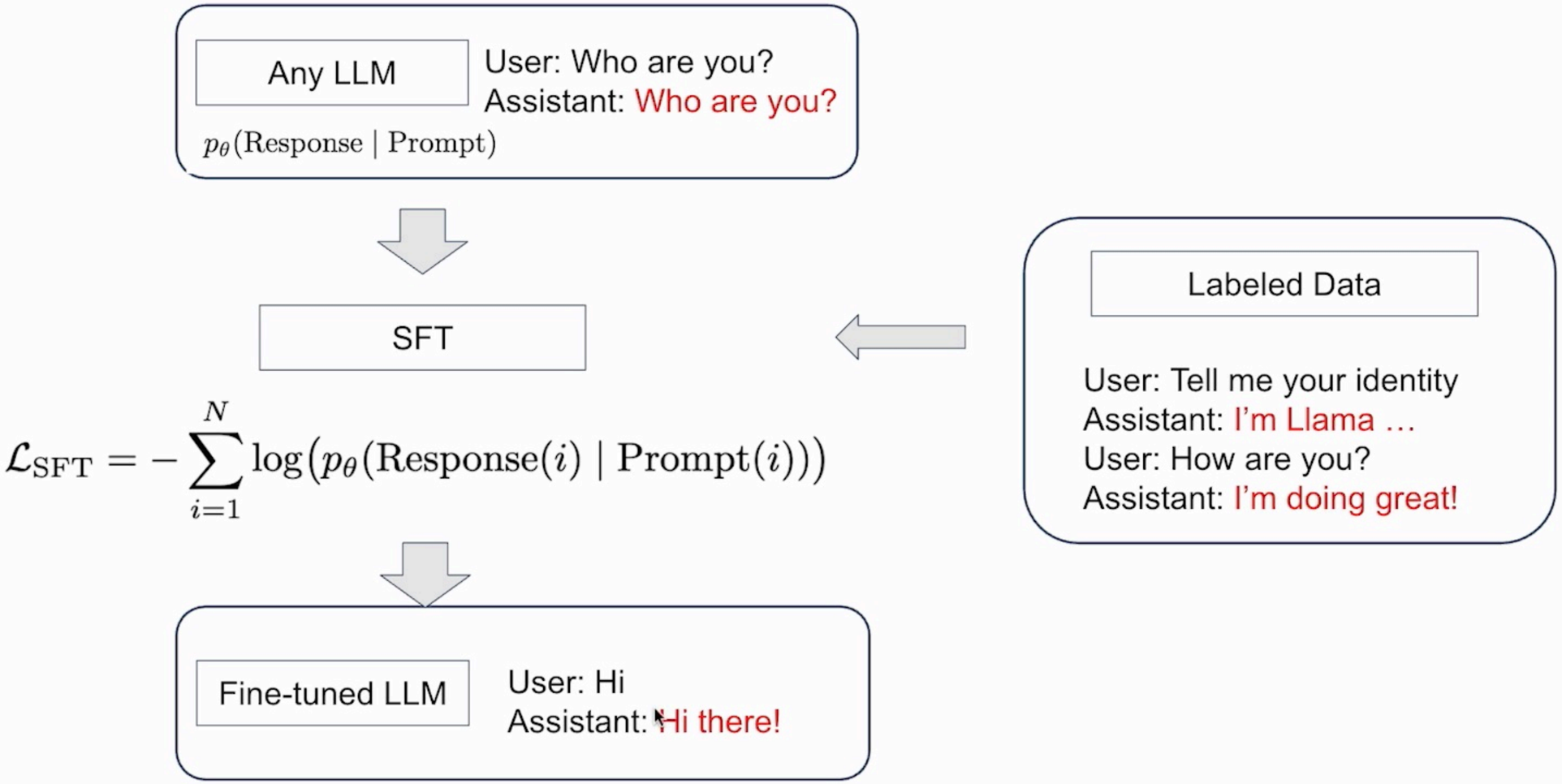
SFT: Imitating Example Responses



SFT: Imitating Example Responses

SFT minimizes negative log likelihood for the responses (maximizes likelihood) with cross entropy loss:

$$\mathcal{L}_{\text{SFT}} = -\sum_{i=1}^{N} \log\left(p_\theta(\text{Response}(i) \mid \text{Prompt}(i))\right)$$

Best Use Cases for SFT

- **Jumpstarting new model behavior**
  - Pre-trained models -> Instruct models
  - Non-reasoning models -> reasoning models
  - Let the model uses certain tools without providing tool descriptions in the prompt

- **Improving model capabilities**
  - Distilling capabilities for small models by training on high-quality synthetic data generated from larger models

Principles of SFT Data Curation

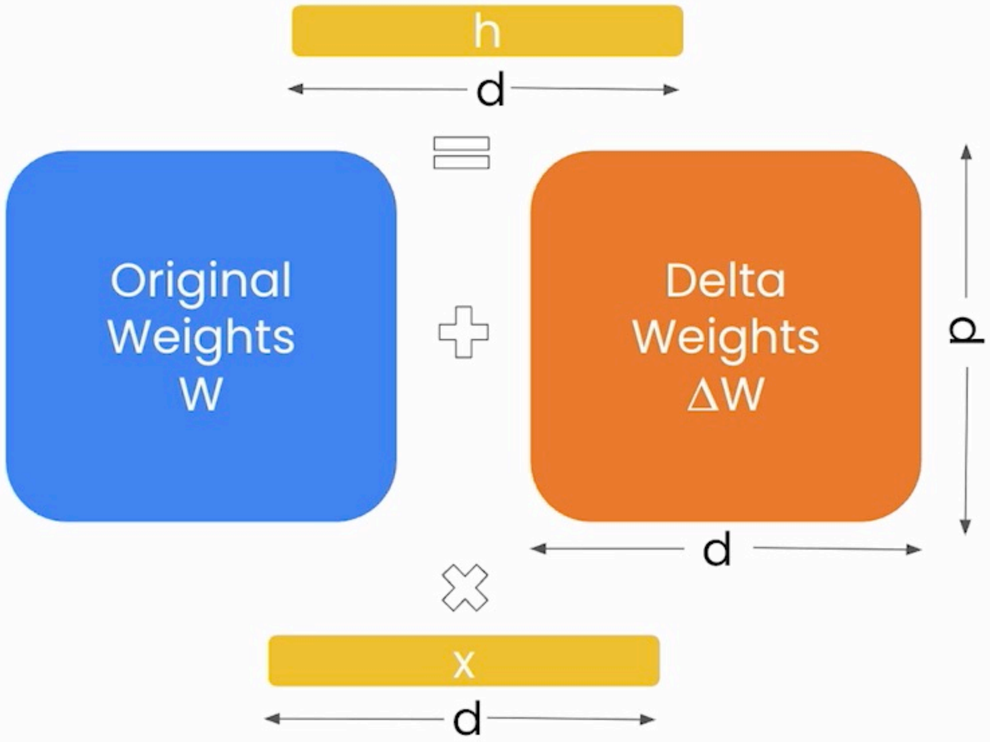
- **Common methods for high-quality SFT data curation:**
  - **Distillation**: Generate responses from a stronger and larger instruct model
  - **Best of K / rejection sampling**: Generate multiple responses from the original model, select the best among them
  - **Filtering**: start from larger scale SFT dataset, filter according to the quality of responses and diversity of the prompts

- **Quality > quantity for improving capabilities:**
  - 1,000 high-quality, diverse data > 1,000,000 mixed-quality data

Full Fine-tuning vs Parameter Efficient Fine-tuning (PEFT)

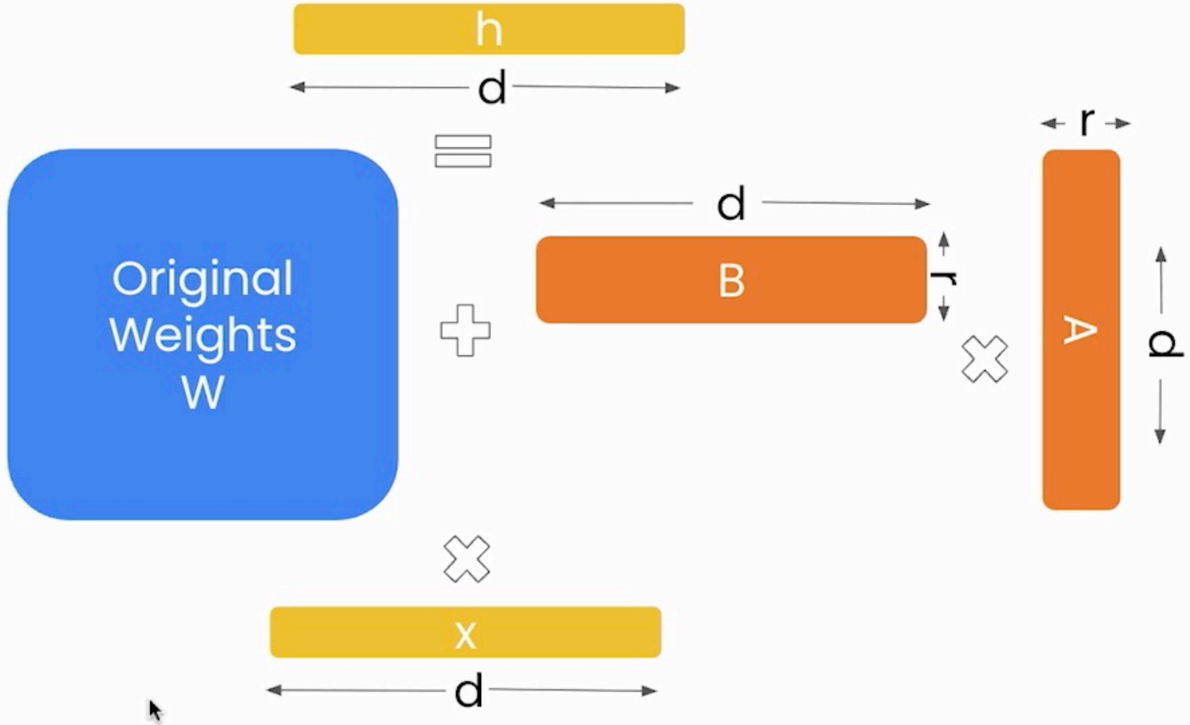
$$h = (W + \Delta W)x$$

$$W, \Delta W \in \mathbb{R}^{d \times d}, h, x \in \mathbb{R}^{d \times 1}$$



$$h = (W + BA)x$$

$$B \in \mathbb{R}^{d \times r}, A \in \mathbb{R}^{r \times d}$$



Both full-finetuning and PEFT can be used in any of the post-training methods.
PEFT like Lora saves memory, learns less while forgets less [1]

[1] Biderman, Dan, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King et al. "Lora learns less and forgets less." *arXiv preprint arXiv:2405.09673* (2024).