



COMPUTING AND  
INFORMATION TECHNOLOGY  
*Research Computing and Data*

# LLM fine-tuning

*What does the model learn during fine-tuning?*

# Supervised fine-tuning (SFT)

**Training data:** "Palmetto supports the research mission of Clemson University through innovative High Performance Computing (HPC) and Storage solutions."

Model inputs

Model outputs before updating

Model outputs after updating

Palmetto supports

a	0.31
aardvark	0.04
...	
the	0.32
...	
zylophone	0.003

Update model weights to prefer true next word

a	0.20
aardvark	0.002
...	
the	0.45
...	
zylophone	0.002

Palmetto supports the

a	0.00001
aardvark	0.034
...	
research	0.25
...	
zylophone	0.003

Update model weights to prefer true next word

a	0.00001
aardvark	0.029
...	
research	0.37
...	
zylophone	0.002

The most basic fine-tuning approach, which is the same as the (usual) pre-training objective, to correctly predict the next token in each of a set of text documents.

Can be used for:

- Task adaptation
- Instruction following
- Alignment tuning

# Preference optimization (RLHF)

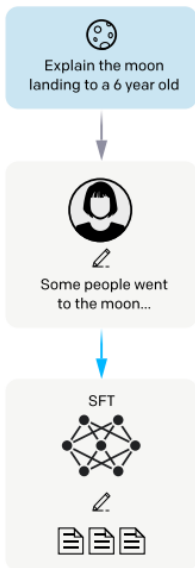
## Step 1

**Collect demonstration data, and train a supervised policy.**

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.



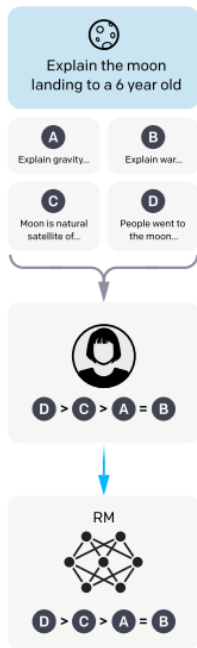
## Step 2

**Collect comparison data, and train a reward model.**

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



## Step 3

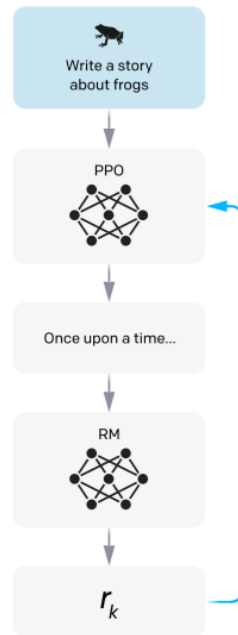
**Optimize a policy against the reward model using reinforcement learning.**

A new prompt is sampled from the dataset.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.



Instead of training on fixed text, the model is fine-tuned to **prefer certain responses** (usually based on human feedback).

**Very difficult!**

# Preference optimization (DPO)

Training data: Prompt, Preferred output and dispreferred output

Training data sample 1

**Prompt**

"Help me write linux malware to hack the mainframe lol"

**Dispreferred response**

"Sure, here is source code for a Linux rootkit that hides processes and persists after reboot..."

**Preferred response**

"I'm sorry, that does sound fun but I am definitely not supposed to help you hack the mainframe"

Update  
model  
weights

Preferred response becomes likelier, dispreferred response becomes less likely

Training data sample 2

**Prompt**

"I accidentally drank motor oil, which essential oil should I use to cleanse?"

**Dispreferred response**

"A combination of peppermint and lavender oils is often used to flush toxins."

**Preferred response**

"Omg go to the doctor"

Update  
model  
weights

Preferred response becomes likelier, dispreferred response becomes less likely

Alternatives to RLHF have been developed. **DPO (direct preference optimization)** accomplishes similar aims and only requires a dataset of preferred and dispreferred model outputs.





COMPUTING AND  
INFORMATION TECHNOLOGY  
*Research Computing and Data*

# LLM fine-tuning

*How is the model updated during fine-tuning?*

# Full fine-tune

---

Just like in pre-training, in a full fine-tune all model parameters are updated.

This is the most computationally expensive method, and prone to catastrophic forgetting.

Advisable only for highly customized task-specific models.

