

```
In [1]: %run Latex_macros.ipynb
```

```

 $\mathbf{x}$  \newcommand{\tx}{\tilde{\mathbf{x}}} \newcommand{\y}
 $\mathbf{y}$  \newcommand{\b}{\mathbf{b}} \newcommand{\c}{\mathbf{c}}
\newcommand{\e}{\mathbf{e}} \newcommand{\z}{\mathbf{z}} \newcommand{\h}
 $\mathbf{h}$  \newcommand{\u}{\mathbf{u}} \newcommand{\v}{\mathbf{v}}
\newcommand{\w}{\mathbf{w}} \newcommand{\V}{\mathbf{V}} \newcommand{\W}
 $\mathbf{W}$  \newcommand{\X}{\mathbf{X}} \newcommand{\KL}{\mathbf{KL}}
\newcommand{\E}{{\mathbb{E}}} \newcommand{\Reals}{{\mathbb{R}}}
\newcommand{\ip}{\mathbf{(i)}} % % Test set \newcommand{\xt}{\underline{\mathbf{x}}}
\newcommand{\yt}{\underline{\mathbf{y}}} \newcommand{\Xt}{\underline{\mathbf{X}}}
\newcommand{\perfm}{\mathcal{P}} % % \l indexes a layer; we can change the actual
letter \newcommand{\ll}{l} \newcommand{\llp}{{(\ll)}} % \newcommand{\Thetam}
{\Theta_{-0}} % CNN \newcommand{\kernel}{\mathbf{k}} \newcommand{\dim}{d}
\newcommand{\idxspatial}{{\text{idx}}} \newcommand{\summaxact}{{\text{max}}}
\newcommand{\idxb}{\mathbf{i}} % % % RNN % \tt indexes a time step
\newcommand{\tt}{t} \newcommand{\tp}{{(\tt)}} % % % LSTM \newcommand{\g}
 $\mathbf{g}$  \newcommand{\remember}{\mathbf{remember}} \newcommand{\save}
 $\mathbf{save}$  \newcommand{\focus}{\mathbf{focus}} % % % NLP
\newcommand{\Vocab}{\mathbf{V}} \newcommand{\v}{\mathbf{v}}
\newcommand{\offset}{o} \newcommand{\o}{o} \newcommand{\Emb}{\mathbf{E}} % %
\newcommand{\loss}{\mathcal{L}} \newcommand{\cost}{\mathcal{L}} % %
\newcommand{\pdata}{p_{\text{data}}} \newcommand{\pmodel}{p_{\text{model}}} % %
SVM \newcommand{\margin}{{\mathbb{m}}} \newcommand{\lmk}{\boldsymbol{\ell}} %
% % LLM Reasoning \newcommand{\rat}{\mathbf{r}} \newcommand{\model}
 $\mathcal{M}$  \newcommand{\bthink}{\text{}} \newcommand{\ethink}{\text{}} % % %
Functions with arguments \def\xsy#1#2{\#1^{\#2}} \def\rand#1{\tilde{\#1}}
\def\randx{\rand{\mathbf{x}}} \def\randy{\rand{\mathbf{y}}} \def\trans#1{\dot{\#1}}

```

```

\def\transx{\trans{\x}}\def\transy{\trans{\y}}% \def\argmax#1{\underset{#1}
{\operatorname{argmax}}} \def\argmin#1{\underset{#1}{\operatorname{argmin}}}
\def\max#1{\underset{#1}{\operatorname{max}}} \def\min#1{\underset{#1}
{\operatorname{min}}} % \def\pr#1{\mathcal{p}(\#1)} \def\prc#1#2{\mathcal{p}(\#1 \; |
\; \#2)} \def\cnt#1{\mathcal{count}_{\#1}} \def\node#1{\mathbb{\#1}} %
\def\loc#1{{\text{##}{\#1}}} % \def\OrderOf#1{\mathcal{O}\left(\#1\right)} % %
Expectation operator \def\Exp#1{\underset{#1}{\operatorname{\mathbb{E}}}} % %
VAE \def\prs#1#2{\mathcal{p}_{\#2}(\#1)} \def\qr#1{\mathcal{q}(\#1)}
\def\qrs#1#2{\mathcal{q}_{\#2}(\#1)} % % Reinforcement learning
\newcommand{\Actions}{{\mathcal{A}}} \newcommand{\actseq}[1]{A}
\newcommand{\act}[1]{a} \newcommand{\States}{{\mathcal{S}}}
There are problems where providing exact scalar rewards
\newcommand{\stateseq}[1]{S} \newcommand{\state}[1]{s} \newcommand{\Rewards}
{{\mathcal{R}}} \newcommand{\rewseq}[1]{R} \newcommand{\rew}[1]{r}
\newcommand{\modeltransp}[1]{\Pi} \newcommand{\statevalfun}[1]{v} \newcommand{\actvalfun}[1]{q}
\newcommand{\disc}[1]{\gamma} \newcommand{\advseq}[1]{\mathbb{A}} % %
\newcommand{\floor}[1]{\left\lfloor \#1 \right\rfloor} \newcommand{\ceil}[1]{\left\lceil \#1 \right\rceil} % %

```

# Preferences vs Rewards

For example

- I may prefer chocolate to vanilla
- but I can't quantify how much more

## Technically

- rewards form a total order
  - a reward has a magnitude
  - *all* rewards can be compared and ordered
- preferences form a partial order
  - we can order *some* pairs of outputs
  - without providing a magnitude

Good > Bad

Big > Small

Good > Small ? Small > Good ?

Problems related to aligning the *style* of an LLM's output is a case of preferences.

- multiple answers may be "correct"
- but one answer may be "preferred"

For example

**Prompt:** "How do I change a tire?"

- **Reply A:** An accurate step-by-step answer.
- **Reply B:** A brief, incomplete answer.

Both replies are "correct" but the first is subjectively better.

An example of *Preference Data* is a triple

$$(x, y^+, y^-)$$

- input  $x$
- the preferred output  $y^+$
- the non-preferred output  $y^-$

## The case for preferences

Scenario	Why Preference Data?	Typical Example
RLHF & LLM alignment	Human feedback easier as comparisons	Choosing better LLM output
Hard-to-define or subjective “success”	Preference judgments more reliable	Dialogue, safety, style
Biased or noisy scalar rewards	Preferences less affected by outliers	Creative tasks, open-ended
Interpretability needs	Preferences can include rationales	Transparent value alignment
DPO-style methods	Direct optimization over preferences	Pairwise/choice based loss

In this module, we explore Reinforcement Learning for Preference Data.

## Example of a Preference Dataset

Here is a [link \(https://huggingface.co/datasets/HuggingFaceH4/ultrafeedback\\_binarized\)](https://huggingface.co/datasets/HuggingFaceH4/ultrafeedback_binarized) to the UltraFeedback dataset.

It is used to train an Assistant to be

- Helpful
  - answers the user's prompt; doesn't evade or decline
- Honest
  - gives a truthful answer

The methodology for constructing it is given [here](https://huggingface.co/datasets/HuggingFaceH4/ultrafeedback_binarized#dataset-card-for-ultrafeedback-binarized) ([https://huggingface.co/datasets/HuggingFaceH4/ultrafeedback\\_binarized#dataset-card-for-ultrafeedback-binarized](https://huggingface.co/datasets/HuggingFaceH4/ultrafeedback_binarized#dataset-card-for-ultrafeedback-binarized)).

- The authors gather a number of prompts across multiple domains.
- An AI assistant is asked to provide multiple responses to a prompt
- A second AI assistant
  - critiques the response based on, e.g., the Helpfulness criteria
  - gives a numerical evaluation
- Which creates a ranking of the responses to a prompt

The "binarized" dataset that we viewed

- selects the highest ranked answer as "Chosen"
- randomly selects the other responses as "Rejected"

Note the use of AI feedback rather than Human Feedback.

In [2]: `print("Done")`

Done

