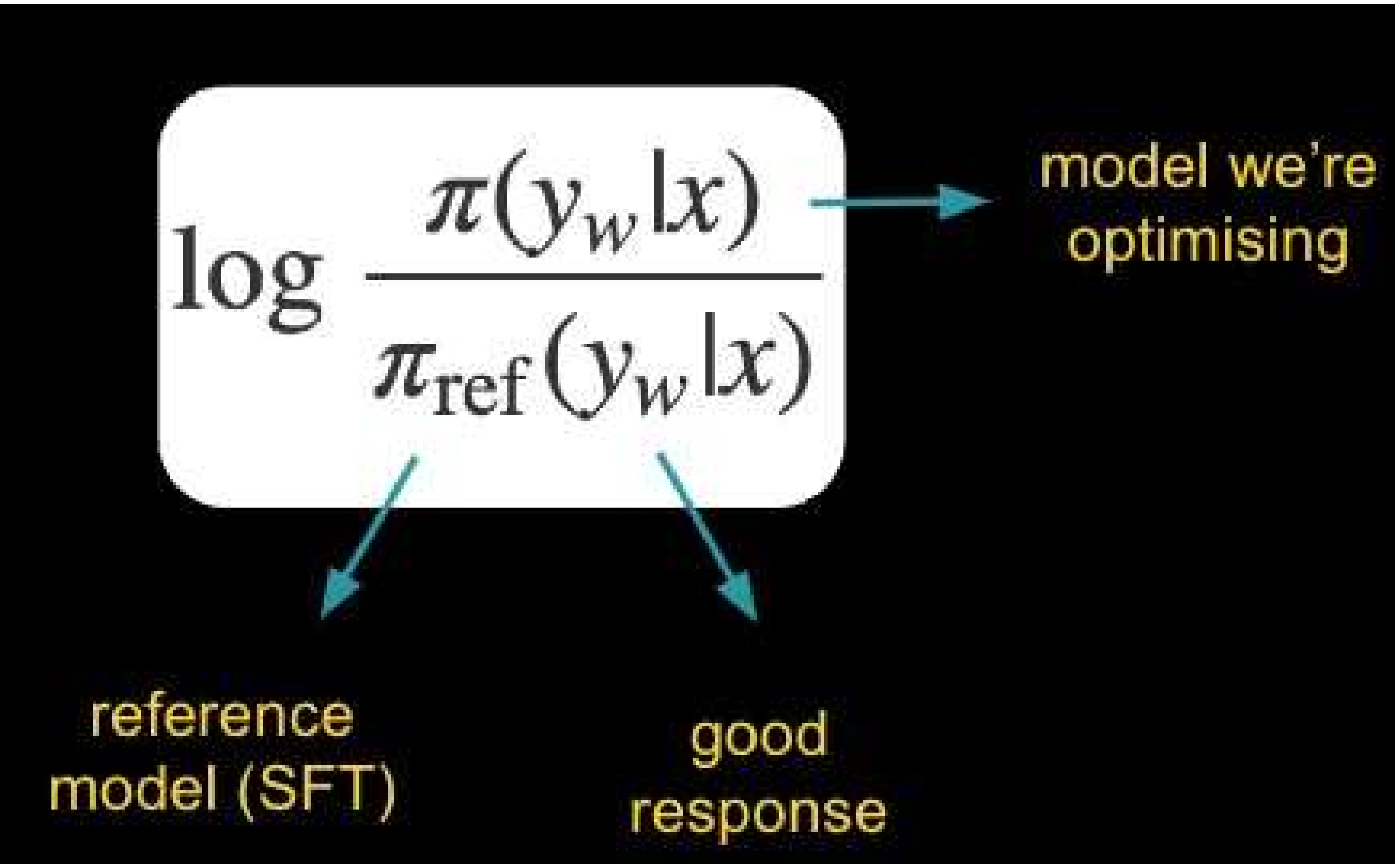
Intuitively DPO uses preference data (given a context/prompt, there is a preferred/good response over a dis-preferred/bad response).

At the heart of DPO is formulation of loss function that considers the likelihood of preferred response over dis-preffered response and optimizes the LLM model towards that objective:

Dataset of preferences $\{(x, yw, yl)\}$, where x is a prompt and yw, yl are the preferred and dis-preferred responses.

$$\max_{\pi} \mathop{\mathbb{E}}_{(x, y_w, y_l) \sim \mathcal{D}} \log \sigma \left(\beta \log \frac{\pi(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \right)$$


$\log \frac{\pi(y_w|x)}{\pi_{\text{ref}}(y_w|x)}$
model we're optimising
reference model (SFT)
good response

DPO outline. The general DPO pipeline is as follows: 1) Sample completions $y_1, y_2 \sim \pi_{\text{ref}}(\cdot \mid x)$ for every prompt x, label with human preferences to construct the offline dataset of preferences $\mathcal{D} = \{x^{(i)}, y_w^{(i)}, y_l)^{(i)}\}_{i=1}^N$ and 2) optimize the language model π_θ to minimize $\mathcal{L}_{\text{DPO}}$ for the given π_{ref} and $\mathcal{D}$ and desired β. In practice, one would like to reuse preference datasets publicly available, rather than generating samples and gathering human preferences. Since the preference datasets are sampled using π^{SFT}, we initialize $\pi_{\text{ref}} = \pi^{\text{SFT}}$ whenever available. However, when π^{SFT} is not available, we initialize π_{ref} by maximizing likelihood of preferred completions (x, y_w), that is, $\pi_{\text{ref}} = \arg\max_\pi \mathbb{E}_{x,y_w\sim\mathcal{D}}\left[\log \pi(y_w \mid x)\right]$. This procedure helps mitigate the distribution shift between the true reference distribution which is unavailable, and π_{ref} used by DPO. Further details related to the implementation and hyperparameters can be found in Appendix B.

What does the DPO update do? For a mechanistic understanding of DPO, it is useful to analyze the gradient of the loss function $\mathcal{L}_{\text{DPO}}$. The gradient with respect to the parameters θ can be written as:

$$\nabla_\theta \mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) =$$
$$-\beta \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\bigg[\underbrace{\sigma(\hat{r}_\theta(x, y_l) - \hat{r}_\theta(x, y_w))}_{\text{higher weight when reward estimate is wrong}}\bigg[\underbrace{\nabla_\theta \log \pi(y_w \mid x)}_{\text{increase likelihood of } y_w} - \underbrace{\nabla_\theta \log \pi(y_l \mid x)}_{\text{decrease likelihood of } y_l}\bigg]\bigg],$$

where $\hat{r}_\theta(x, y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$ is the reward implicitly defined by the language model π_θ and reference model π_{ref} (more in Section 5). Intuitively, the gradient of the loss function $\mathcal{L}_{\text{DPO}}$ increases the likelihood of the preferred completions y_w and decreases the likelihood of dispreferred completions y_l. Importantly, the examples are weighed by how much higher the implicit reward model $\hat{r}_\theta$ rates the dispreferred completions, scaled by β, i.e, how incorrectly the implicit reward model orders the completions, accounting for the strength of the KL constraint. Our experiments suggest the importance of this weighting, as a naïve version of this method without the weighting coefficient can cause the language model to degenerate (Appendix Table 3).