Direct Preference Optimization for Image Generation

Multimodal Generative AI Theories and Applications

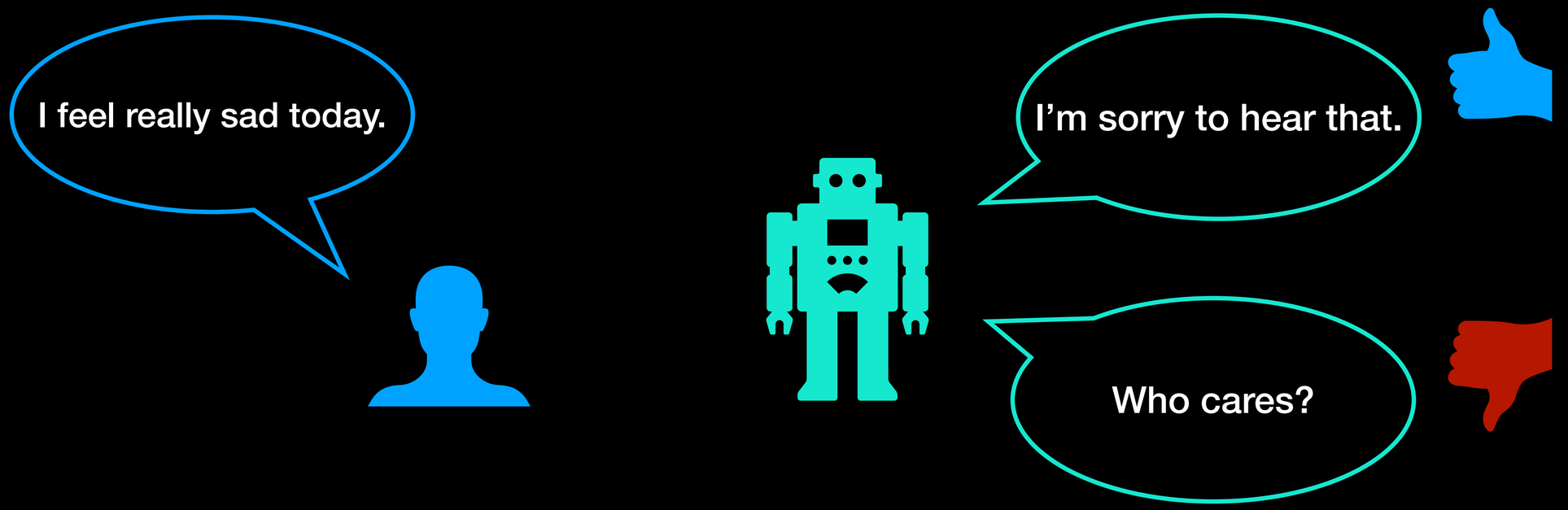
Lecture 10

Gayoung Lee, Jin-Hwa Kim

NAVER AI LAB

Aligning models with human preferences

- We want to fine-tune the model $\hat{p}_\theta$ from the pre-trained model p_θ to generate samples that are preferred by humans.



Preference optimization

- If we have a *reward model* r that predicts human preference, we can optimize the model $\hat{p}_\theta$ using:

Keep the model close to the original model

$$\max_{\hat{p}_\theta} \mathbb{E}_{c\sim\mathcal{D}_c,\, x_0\sim\hat{p}_\theta(x_0|c)} \left[r(c, x_0)\right] - \beta \mathrm{KL}\left[\hat{p}_\theta(x_0 \mid c) \| p_\theta(x_0 \mid c)\right]$$

Increasing human preference

where $\beta > 0$ controls the regularization. c is a conditional input, such as a text prompt.

Sequence tutor: Conservative fine-tuning of sequence generation models with kl-control, Jaques et al,. ICML 2017.

Bradley-Terry model

- How can we get a reward model?
- The Bradley-Terry (BT)* model defines the probability that a human prefers x_0^w over w_0^l as:

$$p_{\mathrm{BT}}(x_0^w \succ x_0^l \mid c) = \sigma\big(r(c, x_0^w) - r(c, x_0^l)\big)$$

where r is the reward model, and x_0^w and x_0^l denote winning and losing samples respectively.

*Rank Analysis of Incomplete Block Designs, Bradley et al,. Biometrika 1952.

Bradley-Terry model

- The reward model r can be parameterized with a neural network ϕ, and learned by maximizing log-likelihood using labeled samples x_0^w and x_0^l:

$$L_{\mathrm{BT}}(\phi) = -\,\mathbb{E}_{c,x_0^w,x_0^l}\left[\log \sigma\left(r_\phi(c, x_0^w) - r_\phi(c, x_0^l)\right)\right]$$

Preference optimization

- A common strategy is to train a separate reward model (*e.g.*, a pairwise preference classifier) and fine-tune the target model using its outputs.
- However, training a separate reward model adds extra complexity and time, and its inaccuracies can negatively affect preference optimization learning.

Intro: Direct Preference Optimization (DPO)

- In the preference optimization equation, we need the reward model:

$$\max_{\hat{p}_\theta} \mathbb{E}_{c\sim\mathcal{D}_c, x_0\sim\hat{p}_\theta(x_0|c)}\left[r(c,x_0)\right] - \beta\,\mathrm{KL}\left[\hat{p}_\theta(x_0 \mid c) \,\|\, p_\theta(x_0 \mid c)\right].$$

- Instead, Direct Preference Optimization (DPO)* proposes a method to directly optimize using human preference data, without requiring a separate reward model.

*Direct Preference Optimization: Your Language Model is Secretly a Reward Model, Rafailov et al., NeuIPS 2023.

Intro: Direct Preference Optimization (DPO)

- *Spoiler:* No need to train a separate reward model r_ϕ. x_0^w and x_0^l are preferred and dispreferred samples, respectively.

$$L_{\text{DPO}}(\theta) = -\,\mathbb{E}_{\mathbf{c},x_0^w,x_0^l}\left[\log\sigma\left(\beta\log\frac{\hat{p}_\theta(x_0^w \mid \mathbf{c})}{p_\theta(x_0^w \mid \mathbf{c})} - \beta\log\frac{\hat{p}_\theta(x_0^l \mid \mathbf{c})}{p_\theta(x_0^l \mid \mathbf{c})}\right)\right]$$

Finding analytic global solution

- In the preference optimization equation, we need a reward model:

$$\max_{\hat{p}_\theta} \mathbb{E}_{c\sim\mathcal{D}_c, x_0\sim\hat{p}_\theta(x_0|c)}\left[r(c,x_0)\right] - \beta\,\mathrm{KL}\left[\hat{p}_\theta(x_0 \mid c)\|p_\theta(x_0 \mid c)\right]$$

- DPO first shows that there is a global solution in the above equation:

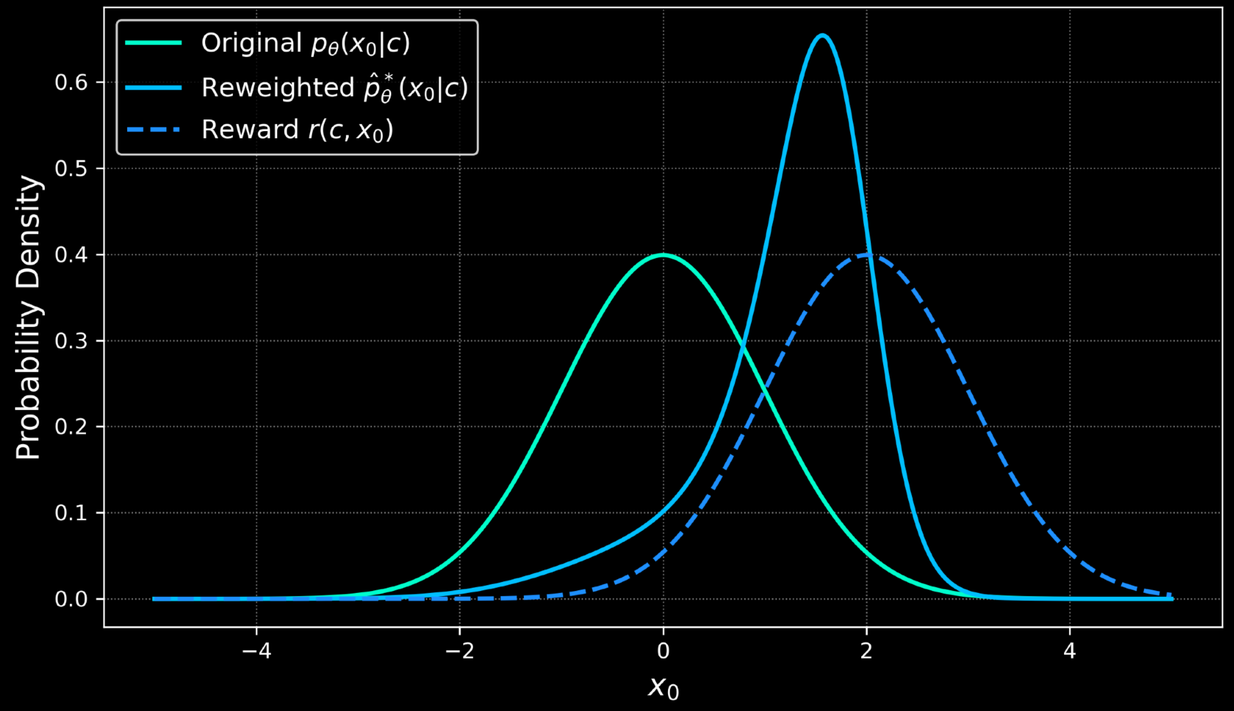
$$\hat{p}^*_\theta(x_0 \mid c) = p_\theta(x_0 \mid c)\exp(r(c,x_0)/\beta)/Z(c)$$

where $Z(c) = \sum_{x_0} p_\theta(x_0 \mid c)\exp(r(c,x_0)/\beta)$ is the partition function.

Direct Preference Optimization: Your Language Model is Secretly a Reward Model, Rafailov et al., NeuIPS 2023.

Visualization of the reweighted distribution

- Visualization of an example: $\hat{p}_\theta^*(x_0 \mid c) = p_\theta(x_0 \mid c)\exp(r(c, x_0)/\beta)/Z(c)$ where $p_\theta \sim \mathcal{N}(0,1)$, reward function $r \sim \mathcal{N}(2,1)$ and $\beta = 0.1$
- We can see that the reward acts as a weighting function.



DPO derivations (1/5)

$$\max \mathbb{E}_{c\sim\mathcal{D},x_0\sim\hat{p}_\theta}\left[r(c,x_0)\right] - \beta\mathrm{KL}\left(\hat{p}_\theta(x_0 \mid c)\|p_\theta(x_0 \mid c)\right) \qquad (\beta > 0)$$

$$= \max \mathbb{E}_{c\sim\mathcal{D}}\mathbb{E}_{x_0\sim\hat{p}_\theta(x_0|c)}\left[r(c,x_0) - \beta\log\frac{\hat{p}_\theta(x_0 \mid c)}{p_\theta(x_0 \mid c)}\right]$$

(KL Divergence definition)

$$\mathrm{KL}(P \parallel Q) = \mathbb{E}_{x\sim P}\left[\log\frac{P(x)}{Q(x)}\right]$$

DPO derivations (2/5)

$$\max \mathbb{E}_{c\sim\mathcal{D},x_0\sim\hat{p}_\theta}\left[r(c,x_0)\right] - \beta\,\mathrm{KL}\left(\hat{p}_\theta(x_0 \mid c)\|p_\theta(x_0 \mid c)\right) \qquad (\beta > 0)$$

$$= \max \mathbb{E}_{c\sim\mathcal{D}}\mathbb{E}_{x_0\sim\hat{p}_\theta(x_0|c)}\left[r(c,x_0) - \beta\log\frac{\hat{p}_\theta(x_0 \mid c)}{p_\theta(x_0 \mid c)}\right] \qquad \text{(KL Divergence definition)}$$

$$= \min \mathbb{E}_{c\sim\mathcal{D}}\mathbb{E}_{x_0\sim\hat{p}_\theta(x_0|c)}\left[\log\frac{\hat{p}_\theta(x_0 \mid c)}{p_\theta(x_0 \mid c)} - \frac{1}{\beta}r(c,x_0)\right] \qquad \textit{(rearrangement)}$$

DPO derivations (3/5)

$$\max \mathbb{E}_{c\sim\mathcal{D},x_0\sim\hat{p}_\theta}\left[r(c,x_0)\right] - \beta\,\mathrm{KL}\left(\hat{p}_\theta(x_0 \mid c)\|p_\theta(x_0 \mid c)\right) \qquad (\beta > 0)$$

$$= \max \mathbb{E}_{c\sim\mathcal{D}}\mathbb{E}_{x_0\sim\hat{p}_\theta(x_0|c)}\left[r(c,x_0) - \beta\log\frac{\hat{p}_\theta(x_0 \mid c)}{p_\theta(x_0 \mid c)}\right] \qquad \text{(KL Divergence definition)}$$

$$= \min \mathbb{E}_{c\sim\mathcal{D}}\mathbb{E}_{x_0\sim\hat{p}_\theta(x_0|c)}\left[\log\frac{\hat{p}_\theta(x_0 \mid c)}{p_\theta(x_0 \mid c)} - \frac{1}{\beta}r(c,x_0)\right] \qquad \text{(rearrangement)}$$

$$= \min \mathbb{E}_{c\sim\mathcal{D}}\mathbb{E}_{x_0\sim\hat{p}_\theta(x_0|c)}\left[\log\frac{\hat{p}_\theta(x_0 \mid c)}{p_\theta(x_0 \mid c)\exp(r(c,x_0)/\beta)}\right] \qquad \textit{(rearrangement)}$$

DPO derivations (3/5)

$$= \min \mathbb{E}_{c\sim\mathcal{D}}\mathbb{E}_{x_0\sim\hat{p}_\theta(x_0|c)}\left[\log \frac{\hat{p}_\theta(x_0 \mid c)}{p_\theta(x_0 \mid c)\exp(r(c,x_0)/\beta)}\right]$$

(rewritten)

$$= \min \mathbb{E}_{c\sim\mathcal{D}}\mathbb{E}_{x_0\sim\hat{p}_\theta(x_0|c)}\left[\log \frac{\hat{p}_\theta(x_0 \mid c)}{\frac{1}{Z(c)}p_\theta(x_0 \mid c)\exp\left(r(c,x_0)/\beta\right)Z(c)}\right]$$

where $Z(c) = \sum_{x_0} p_\theta(x_0 \mid c)\exp\left(r(c,x_0)/\beta\right)$

(introducing $Z(c)$ to make the denominator a valid probability)

DPO derivations (4/5)

$$= \min \mathbb{E}_{c\sim\mathcal{D}}\mathbb{E}_{x_0\sim\hat{p}_\theta(x_0|c)}\left[\log \frac{\hat{p}_\theta(x_0 \mid c)}{\frac{1}{Z(c)}p_\theta(x_0 \mid c)\exp\left(r(c,x_0)/\beta\right)Z(c)}\right]$$

$$= \min \mathbb{E}_{c\sim\mathcal{D}}\left[\mathbb{E}_{x_0\sim\hat{p}_\theta(x_0|c)}\left[\log \frac{\hat{p}_\theta(x_0 \mid c)}{\hat{p}^*_\theta(x_0 \mid c)}\right] - \log Z(c)\right]$$

where $\hat{p}^*_\theta(x_0 \mid c) := \frac{1}{Z(c)}p_\theta(x_0 \mid c)\exp\left(\frac{1}{\beta}r(c,x_0)\right)$

(substitution and rearrangement)

DPO derivations (5/5)

$$= \min \mathbb{E}_{c\sim\mathcal{D}} \left[\mathbb{E}_{x_0\sim\hat{p}_\theta(x_0|c)} \left[\log \frac{\hat{p}_\theta(x_0 \mid c)}{\hat{p}^*_\theta(x_0 \mid c)} \right] - \log Z(c) \right]$$

(rewritten)

$$= \min \mathbb{E}_{c\sim\mathcal{D}} \left[\mathrm{KL}\left(\hat{p}_\theta(x_0 \mid c) \,\|\, \hat{p}^*_\theta(x_0 \mid c)\right) - \log Z(c) \right]$$

- The minimum value of KL-divergence is achieved if and only if two distributions are the same. Therefore, the optimal $\hat{p}_\theta(x_0 \mid c)$ is $\hat{p}^*_\theta(x_0 \mid c)$.

$$\hat{p}^*_\theta(x_0 \mid c) = \frac{1}{Z(c)} p_\theta(x_0 \mid c) \exp\left(\frac{1}{\beta} r(c, x_0)\right)$$

Defining the reward function r

- We derived the optimal solution $\hat{p}_\theta(x_0 \mid c) = p_\theta(x_0 \mid c)\exp\left(r(c, x_0)/\beta\right)/Z(c)$ where

  $Z(c) = \sum_{x_0} p_\theta(x_0 \mid c)\exp\left(r(c, x_0)/\beta\right)$ is the partition function.

- Then, the above equation can be rewritten as:

$$r(c, x_0) = \beta \log \frac{\hat{p}_\theta(x_0 \mid c)}{p_\theta(x_0 \mid c)} + \beta \log Z(c).$$

Plugging reward into BT model

- We can now plug-in the reward function into Bradley-Terry model objective:

$$L_{\text{BT}}(\phi) = -\,\mathbb{E}_{c,x_0^w,x_0^l}\left[\log\sigma\left(r_\phi(c,x_0^w) - r_\phi(c,x_0^l)\right)\right]$$

$$r_\theta(c,x_0) = \beta\log\frac{\hat{p}_\theta(x_0 \mid c)}{p_\theta(x_0 \mid c)} + \beta\log Z(c)$$

$$L_{\text{DPO}}(\theta) = -\,\mathbb{E}_{\mathbf{c},x_0^w,x_0^l}\left[\log\sigma\left(\beta\log\frac{\hat{p}_\theta(x_0^w \mid \mathbf{c})}{p_\theta(x_0^w \mid \mathbf{c})} - \beta\log\frac{\hat{p}_\theta(x_0^l \mid \mathbf{c})}{p_\theta(x_0^l \mid \mathbf{c})}\right)\right]$$

Using softplus for numerical stability.

$$-\log\sigma(z) = -\log\left(\frac{1}{1+e^{-z}}\right) = \log\left(1+e^{-z}\right)$$

DPO summary

$$L_{\text{DPO}}(\theta) = -\mathbb{E}_{\mathbf{c},x_0^w,x_0^l}\left[\log\sigma\left(\beta\log\frac{\hat{p}_\theta(x_0^w \mid \mathbf{c})}{p_\theta(x_0^w \mid \mathbf{c})} - \beta\log\frac{\hat{p}_\theta(x_0^l \mid \mathbf{c})}{p_\theta(x_0^l \mid \mathbf{c})}\right)\right]$$

1. Collect a dataset of paired winning and losing samples.
2. Sample one paired sample, x_0^w and x_0^l.
3. Given a condition $\mathbf{c}$ (*e.g.*, text prompt), compare the likelihood ratio between the pretrained model p_θ and the fine-tuned model $\hat{p}_\theta$.

DPO evaluation

- Human evaluation (collecting preferred responses by human)
- Model evaluation (*e.g.*, asking GPT which answer is better)

	DPO	**SFT**	**PPO-1**
N respondents	272	122	199
GPT-4 (S) win %	47	27	13
GPT-4 (C) win %	54	32	12
Human win %	58	43	17
GPT-4 (S)-H agree	70	77	86
GPT-4 (C)-H agree	67	79	85
H-H agree	65	-	87

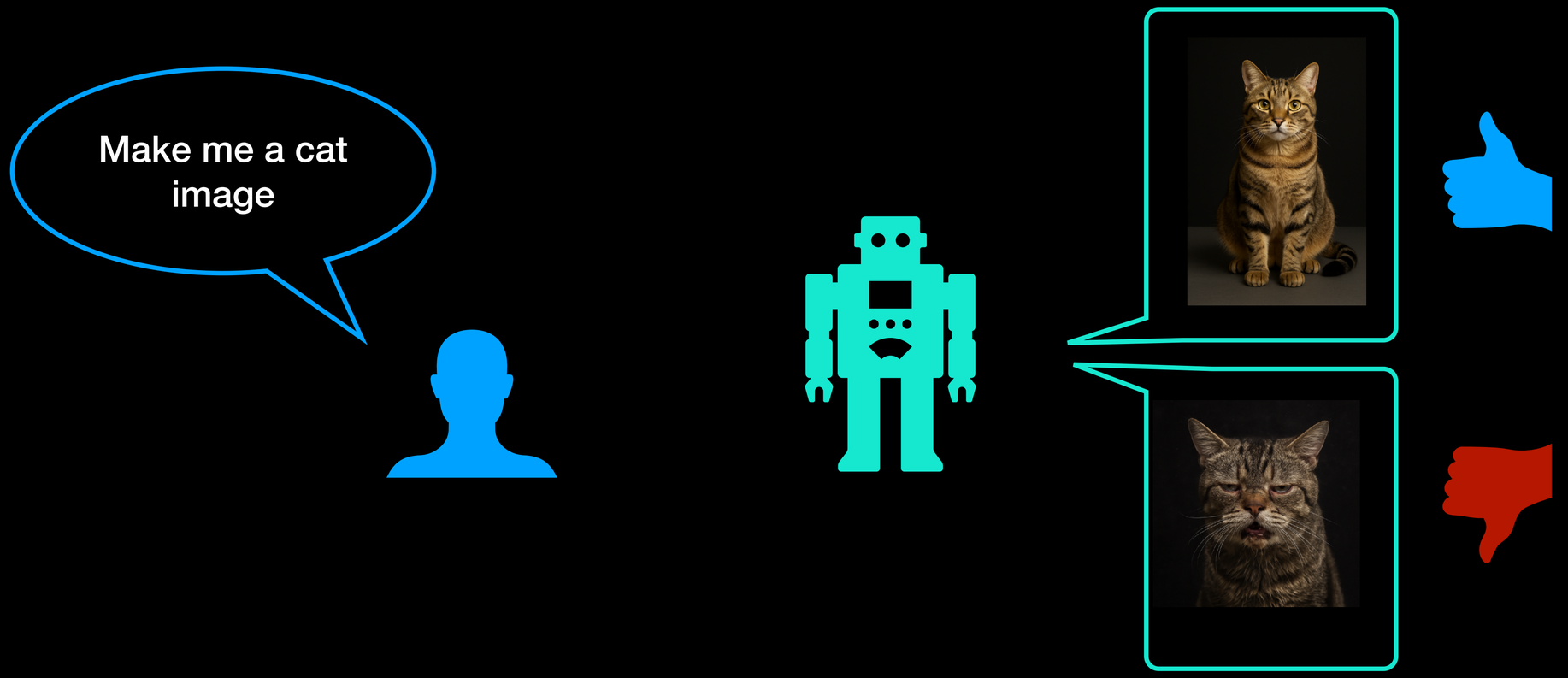
Comparing human and GPT-4 win rates on TL;DR summarization samples*

(S: summary, C: concise, SFT: supervised fine-tuning, PPO: proximal policy optimization)

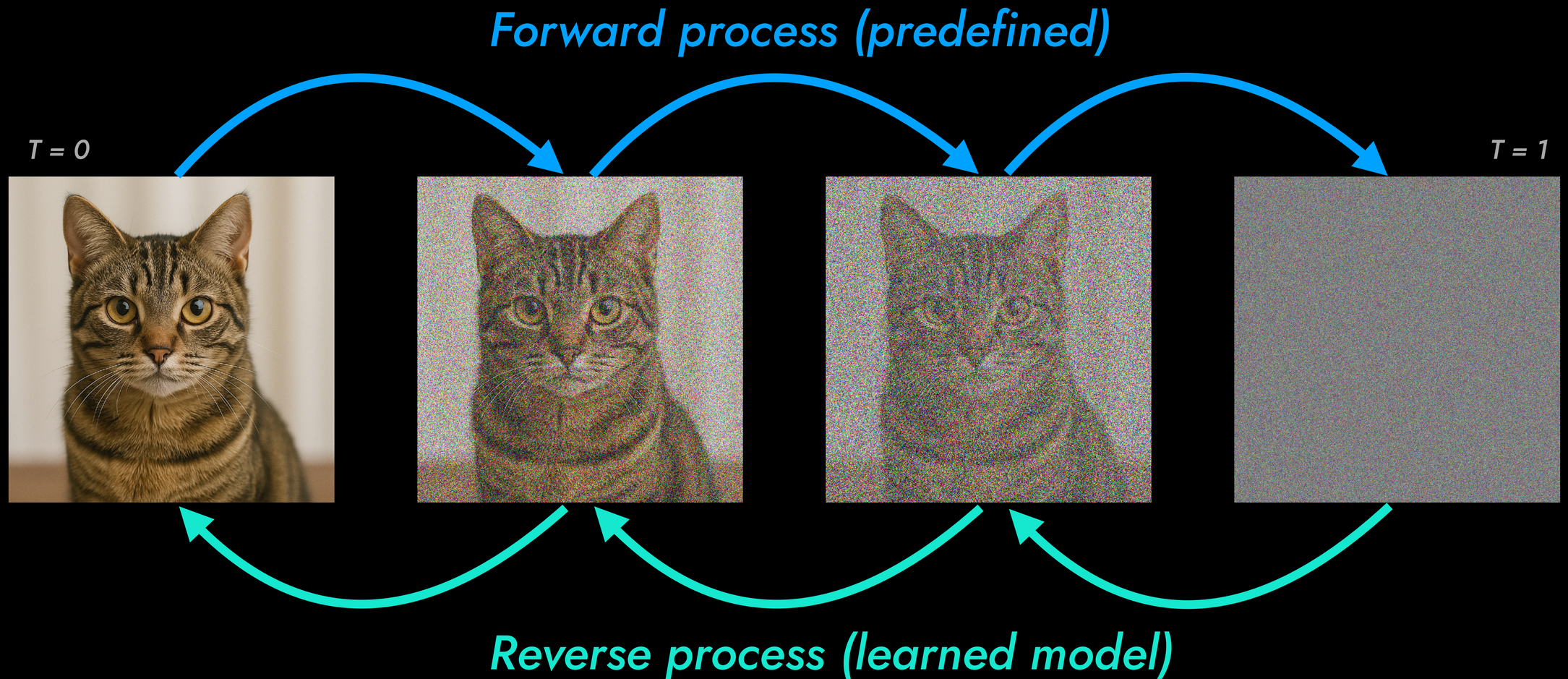
*Direct Preference Optimization: Your Language Model is Secretly a Reward Model, Rafailov et al., NeuIPS 2023.

Preference optimization for images?

- We want to fine-tune the model $\hat{p}_\theta$ from the pre-trained model p_θ to generate samples that are preferred by humans.



Diffusion models (Recap)



Denoising Diffusion Probabilistic Models, Ho et al., NeurIPS 2020.

Diffusion models - DDPM (Recap)

- *Forward process:* Add a noise with a predefined noise schedule.

$$q(x_t \mid x_{t-1}) = \mathcal{N}\left(x_t; \sqrt{1-\beta_t}\, x_{t-1}, \beta_t I\right)$$

$$q(x_t \mid x_0) = \mathcal{N}\left(x_t; \sqrt{\bar{\alpha}_t}\, x_0, (1-\bar{\alpha}_t) I\right)$$

x_0: original data (e.g., clean image)

x_t: noisy data at the timestep t

β_t: predefined variance schedule, $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$

Denoising Diffusion Probabilistic Models, Ho et al., NeurIPS 2020.

Diffusion models - DDPM (Recap)

- *Reverse process:* Remove the noise using a learned model θ.

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}\left(x_{t-1};\, \mu_\theta(x_t, t), \tilde{\beta}_t I\right)$$

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\,\epsilon_\theta(x_t, t)\right)$$

$$L = \mathbb{E}_{x_0,t,\epsilon}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2\right]$$

$\epsilon_\theta(x_t, t)$: Predicted noise, $\epsilon \sim \mathcal{N}(0, I)$, $\tilde{\beta}_t = \dfrac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\,\beta_t$

Denoising Diffusion Probabilistic Models, Ho et al., NeurIPS 2020.

Preference optimization for diffusion model

- Recap the preference optimization

$$\max_{\hat{p}_\theta} \mathbb{E}_{c\sim\mathcal{D}_c, x_0\sim\hat{p}_\theta(x_0|c)} \left[r(c, x_0) \right] - \beta \, \mathrm{KL} \left[\hat{p}_\theta(x_0 \mid c) \,\|\, p_\theta(x_0 \mid c) \right]$$

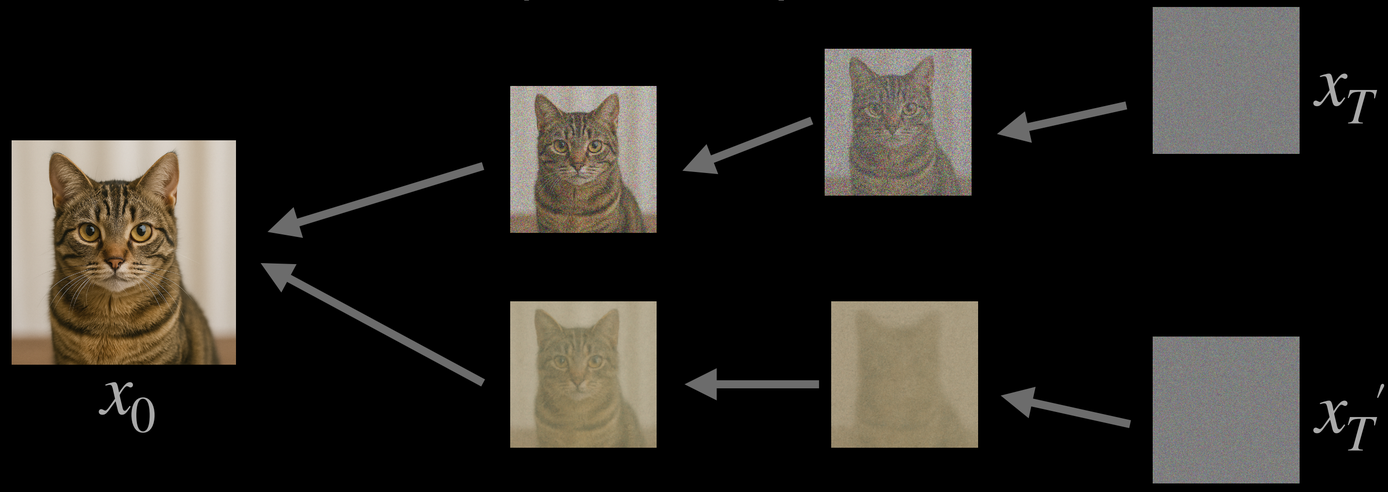
- We want to fine-tune the model $\hat{p}_\theta$ from the pre-trained model p_θ to generate samples that are preferred by humans.

Preference optimization for diffusion model

- Recap the preference optimization

$$\max_{\hat{p}_\theta} \mathbb{E}_{c\sim\mathcal{D}_c, x_0\sim\hat{p}_\theta(x_0|c)}\left[r(c,x_0)\right] - \beta\,\mathrm{KL}\left[\hat{p}_\theta(x_0 \mid c) \,\|\, p_\theta(x_0 \mid c)\right]$$

- To train a diffusion model with $r(c, x_0)$, we would need to consider all possible generation trajectories, which is computationally intractable.



Diffusion-DPO

- Defines a reward function $R(c, x_{0:T})$ for a single trajectory $x_{0:T} = (x_0, x_1, \cdots, x_T)$
- The original reward function is then defined as:

$$r(c, x_0) = \mathbb{E}_{x_{1:T} \sim \hat{p}_\theta(x_{1:T}|x_0,c)} \left[R(c, x_{0:T}) \right]$$

- How can we change the below preference optimization equation with the new reward function $R(c, x_{0:T})$?

$$\max_{\hat{p}_\theta} \mathbb{E}_{c \sim \mathcal{D}_c, x_0 \sim \hat{p}_\theta(x_0|c)} \left[r(c, x_0) \right] - \beta \, \mathrm{KL} \left[\hat{p}_\theta(x_0 \mid c) \,\|\, p_\theta(x_0 \mid c) \right]$$

* Diffusion Model Alignment Using Direct Preference Optimization, Wallace et al., CVPR 2024.

Diffusion-DPO derivations (1/6)

- Start from preference optimization objective:

$$\min_{\hat{p}_\theta} \mathbb{E}_{x_0 \sim \hat{p}_\theta(x_0|c)} \left[-r(c, x_0) \right] + \beta \, \mathrm{KL} \left[\hat{p}_\theta(x_0 \mid c) \,\|\, p_\theta(x_0 \mid c) \right]$$

$$\leq \min_{\hat{p}_\theta} \mathbb{E}_{x_0 \sim \hat{p}_\theta(x_0|c)} \left[-r(c, x_0) \right] + \beta \, \mathrm{KL} \left[\hat{p}_\theta(x_{0:T} \mid c) \,\|\, p_\theta(x_{0:T} \mid c) \right]$$

$$\mathrm{KL} \left[\hat{p}_\theta(x_0 \mid c) \,\|\, p_\theta(x_0 \mid c) \right] \leq \mathrm{KL} \left[\hat{p}_\theta(x_{0:T} \mid c) \,\|\, p_\theta(x_{0:T} \mid c) \right]$$

$\Rightarrow$ Proof on the next slide

Diffusion-DPO derivations (1/6)

$$\mathrm{KL}(\hat{p}_\theta(x_{0:T}) \,\|\, p_\theta(x_{0:T}))$$

$$= \sum_{x_{0:T}} \hat{p}_\theta(x_{0:T}) \log \frac{\hat{p}_\theta(x_{0:T})}{p_\theta(x_{0:T})} \qquad \textit{(definition)}$$

$$= \sum_{x_{0:T}} \hat{p}_\theta(x_{0:T}) \left(\log \frac{\hat{p}_\theta(x_0)}{p_\theta(x_0)} + \log \frac{\hat{p}_\theta(x_{1:T} \mid x_0)}{p_\theta(x_{1:T} \mid x_0)} \right) \qquad \textit{(chain rule)}$$

$$= \sum_{x_0} \hat{p}_\theta(x_0) \sum_{x_{1:T}} \hat{p}_\theta(x_{1:T} \mid x_0) \log \frac{\hat{p}_\theta(x_0)}{p_\theta(x_0)} + \sum_{x_0} \hat{p}_\theta(x_0) \sum_{x_{1:T}} \hat{p}_\theta(x_{1:T} \mid x_0) \log \frac{\hat{p}_\theta(x_{1:T} \mid x_0)}{p_\theta(x_{1:T} \mid x_0)}$$

$$= \sum_{x_0} \hat{p}_\theta(x_0) \log \frac{\hat{p}_\theta(x_0)}{p_\theta(x_0)} + \sum_{x_0} \hat{p}_\theta(x_0) \sum_{x_{1:T}} \hat{p}_\theta(x_{1:T} \mid x_0) \log \frac{\hat{p}_\theta(x_{1:T} \mid x_0)}{p_\theta(x_{1:T} \mid x_0)} \qquad \textit{(chain rule)}$$

Diffusion-DPO derivations (1/6)

$$\mathrm{KL}(\hat{p}_\theta(x_{0:T}) \,\|\, p_\theta(x_{0:T}))$$

$$= \sum_{x_0} \hat{p}_\theta(x_0) \log \frac{\hat{p}_\theta(x_0)}{p_\theta(x_0)} + \sum_{x_0} \hat{p}_\theta(x_0) \sum_{x_{1:T}} \hat{p}_\theta(x_{1:T} \mid x_0) \log \frac{\hat{p}_\theta(x_{1:T} \mid x_0)}{p_\theta(x_{1:T} \mid x_0)}$$

$$= \mathrm{KL}(\hat{p}_\theta(x_0) \,\|\, p_\theta(x_0)) + \mathbb{E}_{x_0 \sim \hat{p}_\theta} \left[\boxed{\mathrm{KL}(\hat{p}_\theta(x_{1:T} \mid x_0) \,\|\, p_\theta(x_{1:T} \mid x_0))} \right]$$

$$\geq 0$$

Diffusion-DPO derivations (2/6)

- We start from preference optimization objective, and omit $\mathbb{E}_{c\sim\mathcal{D}}$ for simplicity).

$$\min_{\hat{p}_\theta} \mathbb{E}_{x_0\sim\hat{p}_\theta(x_0|c)}\left[-r(c,x_0)\right] + \beta\,\mathrm{KL}\left[\hat{p}_\theta(x_0 \mid c)\|p_\theta(x_0 \mid c)\right]$$

$$\leq \min_{\hat{p}_\theta} \mathbb{E}_{x_0\sim\hat{p}_\theta(x_0|c)}\left[-r(c,x_0)\right] + \beta\,\mathrm{KL}\left[\hat{p}_\theta(x_{0:T} \mid c)\|p_\theta(x_{0:T} \mid c)\right]$$

$$= \min_{\hat{p}_\theta} \mathbb{E}_{x_{0:T}\sim\hat{p}_\theta(x_{0:T}|c)}\left[-R(c,x_{0:T})\right] + \beta\,\mathrm{KL}\left[\hat{p}_\theta(x_{0:T} \mid c)\|p_\theta(x_{0:T} \mid c)\right]$$

Remember we defined $r(c,x_0) = \mathbb{E}_{x_{1:T}\sim p_\theta(x_{1:T}|x_0,c)}\left[R(c,x_{0:T})\right]$.

Diffusion-DPO derivations (3/6)

- Start from preference optimization objective:

$$\min_{\hat{p}_\theta} \mathbb{E}_{x_0 \sim \hat{p}_\theta(x_0|c)} \left[-r(c, x_0) \right] + \beta \, \mathrm{KL} \left[\hat{p}_\theta(x_0 \mid c) \| p_\theta(x_0 \mid c) \right]$$

$$\leq \min_{\hat{p}_\theta} \mathbb{E}_{x_0 \sim \hat{p}_\theta(x_0|c)} \left[-r(c, x_0) \right] + \beta \, \mathrm{KL} \left[\hat{p}_\theta(x_{0:T} \mid c) \| p_\theta(x_{0:T} \mid c) \right]$$

$$= \min_{\hat{p}_\theta} \mathbb{E}_{x_{0:T} \sim \hat{p}_\theta(x_{0:T}|c)} \left[-R(c, x_{0:T}) \right] + \beta \, \mathrm{KL} \left[\hat{p}_\theta(x_{0:T} \mid c) \| p_\theta(x_{0:T} \mid c) \right]$$

$$= \min_{\hat{p}_\theta} \mathbb{E}_{x_{0:T} \sim \hat{p}_\theta(x_{0:T}|c)} \left[-R(c, x_{0:T}) \right] + \mathbb{E}_{x_{0:T} \sim \hat{p}_\theta(x_{0:T}|c)} \left[\beta \log \frac{\hat{p}_\theta(x_{0:T} \mid c)}{p_\theta(x_{0:T} \mid c)} \right] \quad \textit{(definition)}$$

$$= \min_{\hat{p}_\theta} \mathbb{E}_{x_{0:T} \sim \hat{p}_\theta(x_{0:T}|c)} \left[-R(c, x_{0:T}) + \beta \log \frac{\hat{p}_\theta(x_{0:T} \mid c)}{p_\theta(x_{0:T} \mid c)} \right] \quad \textit{(rearrangement)}$$

Diffusion-DPO derivations (4/6)

$$= \min_{\hat{p}_\theta} \mathbb{E}_{x_{0:T} \sim \hat{p}_\theta(x_{0:T}|c)} \left[-R(c, x_{0:T}) + \beta \log \frac{\hat{p}_\theta(x_{0:T} \mid c)}{p_\theta(x_{0:T} \mid c)} \right]$$

(rewritten)

$$= \min \mathbb{E}_{x_{0:T} \sim \hat{p}_\theta(x_{0:T}|c)} \left[\log \frac{\hat{p}_\theta(x_{0:T} \mid c)}{\frac{1}{Z(c)} p_\theta(x_{0:T} \mid c) \exp\left(R(c, x_{0:T})/\beta\right) Z(c)} \right]$$

where $Z(c) = \sum_{x_{0:T}} p_\theta(x_{0:T} \mid c) \exp\left(R(c, x_{0:T})/\beta\right)$

(introducing $Z(c)$ to make the denominator a valid probability)

Diffusion-DPO derivations (5/6)

$$= \min \mathbb{E}_{x_{0:T} \sim \hat{p}_\theta(x_{0:T}|c)} \left[\log \frac{\hat{p}_\theta(x_{0:T} \mid c)}{\frac{1}{Z(c)} p_\theta(x_{0:T} \mid c) \exp\left(R(c, x_{0:T})/\beta\right) Z(c)} \right]$$

(rewritten)

$$= \min \mathbb{E}_{x_{0:T} \sim \hat{p}_\theta(x_{0:T}|c)} \left[\log \frac{\hat{p}_\theta(x_{0:T} \mid c)}{\hat{p}^*_\theta(x_{0:T} \mid c)} - \log Z(c) \right]$$

where $\hat{p}^*_\theta(x_{0:T} \mid c) := \frac{1}{Z(c)} p_\theta(x_{0:T} \mid c) \exp\left(\frac{1}{\beta} R(c, x_{0:T})\right)$

(substitution and rearrangement)

Diffusion-DPO derivations (6/6)

$$= \min \mathbb{E}_{x_{0:T} \sim \hat{p}_\theta(x_{0:T}|c)} \left[\log \frac{\hat{p}_\theta(x_{0:T} \mid c)}{\hat{p}^*_\theta(x_{0:T} \mid c)} \right] - \log Z(c) \qquad \textit{(rewritten)}$$

$$= \min \mathrm{KL}\left(\hat{p}_\theta(x_{0:T} \mid c) \parallel \hat{p}^*_\theta(x_{0:T} \mid c)\right) - \log Z(c) \qquad \textit{(by KL Divergence definition)}$$

- The minimum value of KL-divergence is achieved if and only if two distributions are the same. Therefore, the optimal $\hat{p}_\theta(x_{0:T} \mid c)$ is $\hat{p}^*_\theta(x_{0:T} \mid c)$.

$$\hat{p}^*_\theta(x_{0:T} \mid c) = \frac{1}{Z(c)} p_\theta(x_{0:T} \mid c) \exp\left(\frac{1}{\beta} R(c, x_{0:T})\right)$$

Derivations to find r

- We can rearrange the equation to get a reward function.

$$\hat{p}^*_\theta(x_{0:T} \mid c) = \frac{1}{Z(c)} p_\theta(x_{0:T} \mid c) \exp\left(\frac{1}{\beta} R(c, x_{0:T})\right)$$

$$\Rightarrow Z(c) \frac{\hat{p}^*_\theta(x_{0:T} \mid c)}{p_\theta(x_{0:T} \mid c)} = \exp\left(\frac{1}{\beta} R(c, x_{0:T})\right)$$

$$\Rightarrow \beta \log\left(Z(c) \frac{\hat{p}^*_\theta(x_{0:T} \mid c)}{p_\theta(x_{0:T} \mid c)}\right) = R(c, x_{0:T})$$

(Plug into the equation $r(c, x_0) = \mathbb{E}_{x_{1:T} \sim p_\theta(x_{1:T} \mid x_0, c)}\left[R(c, x_{0:T})\right]$*)*

$$\Rightarrow r(c, x_0) = \beta \mathbb{E}_{x_{1:T} \sim \hat{p}_\theta(x_{1:T} \mid x_0, c)} \log \frac{\hat{p}_\theta(x_{0:T} \mid c)}{p_\theta(x_{0:T} \mid c)} + \beta \log Z(c)$$

Derivation of the loss (1/13)

- Now, we can now plug-in the reward r into the Bradley-Terry model objective:

$$L_{\text{BT}}(\phi) = -\,\mathbb{E}_{c,x_0^w,x_0^l}\left[\log\sigma\left(r_\phi(c,x_0^w) - r_\phi(c,x_0^l)\right)\right]$$

$$\text{where } r(c,x_0) = \beta\mathbb{E}_{x_{1:T}\sim\hat{p}_\theta(x_{1:T}|x_0,c)}\log\frac{\hat{p}_\theta(x_{0:T}\mid c)}{p_\theta(x_{0:T}\mid c)} + \beta\log Z(c)$$

$$L_{\text{Diffusion-DPO}}(\theta) = -\,\mathbb{E}_{x_0^w,\,x_0^l\,\sim\,\mathcal{D}}\log\sigma\left(\beta\mathbb{E}_{\substack{x_{1:T}^w\,\sim\,\hat{p}_\theta(x_{1:T}^w\mid x_0^w)\\ x_{1:T}^l\,\sim\,\hat{p}_\theta(x_{1:T}^l\mid x_0^l)}}\left[\log\frac{\hat{p}_\theta(x_{0:T}^w)}{p_\theta(x_{0:T}^w)} - \log\frac{\hat{p}_\theta(x_{0:T}^l)}{p_\theta(x_{0:T}^l)}\right]\right)$$

Derivation of the loss (2/13)

- We now have a loss function for the Diffusion-DPO:

$$L_{\text{Diffusion-DPO}}(\theta) = -\mathbb{E}_{x_0^w, x_0^l \sim \mathcal{D}} \log \sigma \left(\beta \mathbb{E}_{\substack{x_{1:T}^w \sim \hat{p}_\theta(x_{1:T}^w \mid x_0^w) \\ x_{1:T}^l \sim \hat{p}_\theta(x_{1:T}^l \mid x_0^l)}} \left[\log \frac{\hat{p}_\theta(x_{0:T}^w)}{p_\theta(x_{0:T}^w)} - \log \frac{\hat{p}_\theta(x_{0:T}^l)}{p_\theta(x_{0:T}^l)} \right] \right).$$

However, we need to sample $x_{1:T} \sim \hat{p}_\theta(x_{1:T} \mid x_0)$ which is both inefficient and intractable. Instead, we utilize the forward process $q(x_{1:T}|x_0)$ for approximation.

(Difference) $p_\theta(x_{1:T}|x_0) = p_\theta(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t)$ is a model trajectory, while $q(x_{1:T}|x_0)$ is simple fixed forward process.

Derivation of the loss (3/13)

$$L_{\text{Diffusion-DPO}}(\theta) = -\mathbb{E}_{x_0^w, x_0^l \sim \mathcal{D}} \log \sigma \left(\beta \mathbb{E}_{\substack{x_{1:T}^w \sim \hat{p}_\theta(x_{1:T}^w \mid x_0^w) \\ x_{1:T}^l \sim \hat{p}_\theta(x_{1:T}^l \mid x_0^l)}} \left[\log \frac{\hat{p}_\theta(x_{0:T}^w)}{p_\theta(x_{0:T}^w)} - \log \frac{\hat{p}_\theta(x_{0:T}^l)}{p_\theta(x_{0:T}^l)} \right] \right)$$

$$\approx -\mathbb{E}_{x_0^w, x_0^l \sim \mathcal{D}} \log \sigma \left(\beta \mathbb{E}_{\substack{x_{1:T}^w \sim q(x_{1:T}^w \mid x_0^w) \\ x_{1:T}^l \sim q(x_{1:T}^l \mid x_0^l)}} \left[\log \frac{\hat{p}_\theta(x_{0:T}^w)}{p_\theta(x_{0:T}^w)} - \log \frac{\hat{p}_\theta(x_{0:T}^l)}{p_\theta(x_{0:T}^l)} \right] \right)$$

Here, x_0^w and x_0^l are sampled from data. $x_{1:T}^w$ and $x_{1:T}^l$ are obtained by simply adding Gaussian noises through the forward process.

Derivation of the loss (4/13)

$$-\log \sigma\left(\beta \, \mathbb{E}_{\substack{x^w_{1:T} \sim q(x^w_{1:T} \mid x^w_0) \\ x^l_{1:T} \sim q(x^l_{1:T} \mid x^l_0)}} \left[\log \frac{\hat{p}_\theta(x^w_{0:T})}{p_\theta(x^w_{0:T})} - \log \frac{\hat{p}_\theta(x^l_{0:T})}{p_\theta(x^l_{0:T})}\right]\right)$$

(rewritten)

$$= -\log \sigma\left(\beta \, \mathbb{E}_{\substack{x^w_{1:T} \sim q(x^w_{1:T} \mid x^w_0) \\ x^l_{1:T} \sim q(x^l_{1:T} \mid x^l_0)}} \left[\sum_{t=1}^{T} \left(\log \frac{\hat{p}_\theta(x^w_{t-1} \mid x^w_t)}{p_\theta(x^w_{t-1} \mid x^w_t)} - \log \frac{\hat{p}_\theta(x^l_{t-1} \mid x^l_t)}{p_\theta(x^l_{t-1} \mid x^l_t)}\right)\right]\right)$$

(using chain rule, markov process)

$$p(x_{0:T}) = p(x_T) \cdot p(x_{T-1} \mid x_T) \cdot p(x_{T-2} \mid x_{T-1}) \;\cdots\; p(x_0 \mid p_1)$$

Derivation of the loss (5/13)

$$-\log\sigma\left(\beta\,\mathbb{E}_{\substack{x^w_{1:T}\sim q(x^w_{1:T}\mid x^w_0)\\ x^l_{1:T}\sim q(x^l_{1:T}\mid x^l_0)}}\left[\log\frac{\hat{p}_\theta(x^w_{0:T})}{p_\theta(x^w_{0:T})}-\log\frac{\hat{p}_\theta(x^l_{0:T})}{p_\theta(x^l_{0:T})}\right]\right)$$

$$=-\log\sigma\left(\beta\,\mathbb{E}_{\substack{x^w_{1:T}\sim q(x^w_{1:T}\mid x^w_0)\\ x^l_{1:T}\sim q(x^l_{1:T}\mid x^l_0)}}\left[\sum_{t=1}^{T}\left(\log\frac{\hat{p}_\theta(x^w_{t-1}\mid x^w_t)}{p_\theta(x^w_{t-1}\mid x^w_t)}-\log\frac{\hat{p}_\theta(x^l_{t-1}\mid x^l_t)}{p_\theta(x^l_{t-1}\mid x^l_t)}\right)\right]\right)$$

$$=-\log\sigma\left(\beta\sum_{t=1}^{T}\mathbb{E}_{\substack{x^w_{1:T}\sim q(x^w_{1:T}\mid x^w_0)\\ x^l_{1:T}\sim q(x^l_{1:T}\mid x^l_0)}}\left[\log\frac{\hat{p}_\theta(x^w_{t-1}\mid x^w_t)}{p_\theta(x^w_{t-1}\mid x^w_t)}-\log\frac{\hat{p}_\theta(x^l_{t-1}\mid x^l_t)}{p_\theta(x^l_{t-1}\mid x^l_t)}\right]\right)$$

(Linearity of expectation: $\mathbb{E}(A+B)=\mathbb{E}(A)+\mathbb{E}(B)$)

Derivation of the loss (6/13)

$$-\log\sigma\left(\beta\sum_{t=1}^{T}\mathbb{E}_{\substack{x^w_{1:T}\sim q(x^w_{1:T}\mid x^w_0)\\ x^l_{1:T}\sim q(x^l_{1:T}\mid x^l_0)}}\left[\log\frac{\hat{p}_\theta(x^w_{t-1}\mid x^w_t)}{p_\theta(x^w_{t-1}\mid x^w_t)}-\log\frac{\hat{p}_\theta(x^l_{t-1}\mid x^l_t)}{p_\theta(x^l_{t-1}\mid x^l_t)}\right]\right) \qquad \textit{(rewritten)}$$

$$=-\log\sigma\left(\beta T\,\mathbb{E}_t\,\mathbb{E}_{\substack{x^w_{t-1,t}\sim q(x_{t-1,t}\mid x^w_0)\\ x^l_{t-1,t}\sim q(x_{t-1,t}\mid x^l_0)}}\left[\log\frac{\hat{p}_\theta(x^w_{t-1}\mid x^w_t)}{p_\theta(x^w_{t-1}\mid x^w_t)}-\log\frac{\hat{p}_\theta(x^l_{t-1}\mid x^l_t)}{p_\theta(x^l_{t-1}\mid x^l_t)}\right]\right)$$

We can marginalize out unused latents from $x_{0:T}$, keeping only $x_{t-1,t}=(x_{t-1},x_t)$.

$$\text{c.f. } \mathbb{E}_{x,y}[f(x)]=\int f(x)\,p(x,y)\,dx\,dy==\int f(x)\,dx\,p(x,y)\,dy=\int f(x)\,p(x)\,dx=\mathbb{E}_x[f(x)]$$

Derivation of the loss (7/13)

$$-\log\sigma\left(\beta\sum_{t=1}^{T}\mathbb{E}_{\substack{x^w_{1:T}\sim q(x^w_{1:T}\mid x^w_0)\\ x^l_{1:T}\sim q(x^l_{1:T}\mid x^l_0)}}\left[\log\frac{\hat{p}_\theta(x^w_{t-1}\mid x^w_t)}{p_\theta(x^w_{t-1}\mid x^w_t)}-\log\frac{\hat{p}_\theta(x^l_{t-1}\mid x^l_t)}{p_\theta(x^l_{t-1}\mid x^l_t)}\right]\right)$$

$$=-\log\sigma\left(\beta T\,\mathbb{E}_t\,\mathbb{E}_{\substack{x^w_{t-1,t}\sim q(x_{t-1,t}\mid x^w_0)\\ x^l_{t-1,t}\sim q(x_{t-1,t}\mid x^l_0)}}\left[\log\frac{\hat{p}_\theta(x^w_{t-1}\mid x^w_t)}{p_\theta(x^w_{t-1}\mid x^w_t)}-\log\frac{\hat{p}_\theta(x^l_{t-1}\mid x^l_t)}{p_\theta(x^l_{t-1}\mid x^l_t)}\right]\right)$$

$$=-\log\sigma\left(\beta T\,\mathbb{E}_{t,\,x^w_t\sim q(x_t\mid x^w_0)\,x^l_t\sim q(x_t\mid x^l_0)}\mathbb{E}_{\substack{x^w_{t-1}\sim q(x_{t-1}\mid x^w_t,x^w_0)\\ x^l_{t-1}\sim q(x_{t-1}\mid x^l_t,x^l_0)}}\left[\log\frac{\hat{p}_\theta(x^w_{t-1}\mid x^w_t)}{p_\theta(x^w_{t-1}\mid x^w_t)}-\log\frac{\hat{p}_\theta(x^l_{t-1}\mid x^l_t)}{p_\theta(x^l_{t-1}\mid x^l_t)}\right]\right)$$

Diffusion-DPO (Derivation of the loss)

$$-\log\sigma\left(\beta T\,\mathbb{E}_{\substack{t,\,x_t^w\sim q(x_t\mid x_0^w)\\ x_t^l\sim q(x_t\mid x_0^l)}}\mathbb{E}_{\substack{x_{t-1}^w\sim q(x_{t-1}\mid x_t^w,x_0^w)\\ x_{t-1}^l\sim q(x_{t-1}\mid x_t^l,x_0^l)}}\left[\log\frac{\hat{p}_\theta(x_{t-1}^w\mid x_t^w)}{p_\theta(x_{t-1}^w\mid x_t^w)}-\log\frac{\hat{p}_\theta(x_{t-1}^l\mid x_t^l)}{p_\theta(x_{t-1}^l\mid x_t^l)}\right]\right)$$

$$\leq -\,\mathbb{E}_{\substack{t,\,x_t^w\sim q(x_t\mid x_0^w)\\ x_t^l\sim q(x_t\mid x_0^l)}}\log\sigma\left(\beta T\mathbb{E}_{\substack{x_{t-1}^w\sim q(x_{t-1}\mid x_t^w,x_0^w)\\ x_{t-1}^l\sim q(x_{t-1}\mid x_t^l,x_0^l)}}\left[\log\frac{\hat{p}_\theta(x_{t-1}^w\mid x_t^w)}{p_\theta(x_{t-1}^w\mid x_t^w)}-\log\frac{\hat{p}_\theta(x_{t-1}^l\mid x_t^l)}{p_\theta(x_{t-1}^l\mid x_t^l)}\right]\right)$$

By *Jensen's inequality* with the convexity of $-log$: $\varphi\left(\mathbb{E}[X]\right)\leq\mathbb{E}\left[\varphi(X)\right]$ if φ is a convex function.

Diffusion-DPO (Derivation of the loss)

$$-\mathbb{E}_{\substack{t,\, x_t^w \sim q(x_t \mid x_0^w) \\ x_t^l \sim q(x_t \mid x_0^l)}} \log \sigma \left(\beta T \mathbb{E}_{\substack{x_{t-1}^w \sim q(x_{t-1} \mid x_t^w, x_0^w) \\ x_{t-1}^l \sim q(x_{t-1} \mid x_t^l, x_0^l)}} \left[\log \frac{\hat{p}_\theta(x_{t-1}^w \mid x_t^w)}{p_\theta(x_{t-1}^w \mid x_t^w)} - \log \frac{\hat{p}_\theta(x_{t-1}^l \mid x_t^l)}{p_\theta(x_{t-1}^l \mid x_t^l)} \right] \right)$$

We can write the above equation using KL-divergence:

$$-\mathbb{E}_{\substack{t,\, x_t^w \sim q(x_t \mid x_0^w) \\ x_t^l \sim q(x_t \mid x_0^l)}} \log \sigma \left(\begin{array}{c} \beta T \big(\mathrm{KL}(q(x_{t-1}^w \mid x_{0,t}^w) \,\|\, \hat{p}_\theta(x_{t-1}^w \mid x_t^w)) - \mathrm{KL}(q(x_{t-1}^w \mid x_{0,t}^w) \,\|\, p_\theta(x_{t-1}^w \mid x_t^w)) \\ -\mathrm{KL}(q(x_{t-1}^l \mid x_{0,t}^l) \,\|\, \hat{p}_\theta(x_{t-1}^l \mid x_t^l)) + \mathrm{KL}(q(x_{t-1}^l \mid x_{0,t}^l) \,\|\, p_\theta(x_{t-1}^l \mid x_t^l)) \big) \end{array} \right)$$

$$\mathrm{KL}(P \,\|\, Q) = \mathbb{E}_{x \sim P} \left[\log \frac{P(x)}{Q(x)} \right]$$

Diffusion-DPO (Derivation of the loss)

$$-\mathbb{E}_{\substack{t,\, x_t^w \sim q(x_t \mid x_0^w) \\ x_t^l \sim q(x_t \mid x_0^l)}} \log \sigma \left(\begin{array}{c} \beta T \Big(\mathrm{KL}(q(x_{t-1}^w \mid x_{0,t}^w) \,\|\, \hat{p}_\theta(x_{t-1}^w \mid x_t^w)) + \mathrm{KL}(q(x_{t-1}^w \mid x_{0,t}^w) \,\|\, p_\theta(x_{t-1}^w \mid x_t^w)) \\ - \mathrm{KL}(q(x_{t-1}^l \mid x_{0,t}^l) \,\|\, \hat{p}_\theta(x_{t-1}^l \mid x_t^l)) + \mathrm{KL}(q(x_{t-1}^l \mid x_{0,t}^l) \,\|\, p_\theta(x_{t-1}^l \mid x_t^l)) \Big) \end{array} \right)$$

Recap the diffusion model:

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}\left(x_{t-1};\, \mu_\theta(x_t, t), \tilde{\beta}_t I\right)$$

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right)$$

KL Divergence of Gaussian distributions:

$$D_{\mathrm{KL}}\left(\mathcal{N}(\mu_0, \Sigma_0) \,\|\, \mathcal{N}(\mu_1, \Sigma_1)\right) = \frac{1}{2}\left(\mathrm{tr}\left(\Sigma_1^{-1}\Sigma_0\right) + (\mu_1 - \mu_0)^\top \Sigma_1^{-1}(\mu_1 - \mu_0) - d + \ln \frac{|\Sigma_1|}{|\Sigma_0|} \right)$$

Diffusion-DPO (Derivation of the loss)

We want to calculate $\mathrm{KL}\left(q(x_{t-1}^w \mid x_{0,t}^w) \,\|\, \hat{p}_\theta(x_{t-1}^w \mid x_t^w)\right)$

Recap the diffusion model

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}\left(x_{t-1};\, \mu_\theta(x_t, t), \tilde{\beta}_t I\right) \qquad \mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\,\epsilon_\theta(x_t, t)\right)$$

And can calculate $q(x_{t-1} \mid x_{0,t})$ since the forward process has a closed form.

$$q(x_{t-1} \mid x_t, x_0) = \mathcal{N}\left(x_{t-1};\, \mu_q(x_t, x_0), \tilde{\beta}_t I\right)$$

$$\mu_q(x_t, x_0) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\,\epsilon\right), \quad \epsilon \sim \mathcal{N}(0, I)$$

Diffusion-DPO (Derivation of the loss)

$$D_{\mathrm{KL}}\left(\mathcal{N}(\mu_0,\Sigma_0)\parallel\mathcal{N}(\mu_1,\Sigma_1)\right)=\frac{1}{2}\left(\mathrm{tr}\left(\Sigma_1^{-1}\Sigma_0\right)+(\mu_1-\mu_0)^{\top}\Sigma_1^{-1}(\mu_1-\mu_0)-d+\ln\frac{|\Sigma_1|}{|\Sigma_0|}\right)$$

In our diffusion model, the means are:

$$\mu_\theta(x_t,t)=\frac{1}{\sqrt{\alpha_t}}\left(x_t-\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\,\epsilon_\theta(x_t,t)\right),\quad \mu_q(x_t,x_0)=\frac{1}{\sqrt{\alpha_t}}\left(x_t-\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\,\epsilon\right).$$

The variances of both distribution are fixed as $\tilde{\beta}_t I$. We can simplify the divergence as:

$$\mathrm{KL}\left(q(x_{t-1}^w\mid x_{0,t}^w)\parallel\hat{p}_\theta(x_{t-1}^w\mid x_t^w)\right)\propto\|\epsilon^w-\epsilon_\theta(x_t^w,t)\|.$$

Diffusion-DPO (Derivation of the loss)

$$L_{\text{DPO-Diffusion}}(\theta) \leq$$

$$-\mathbb{E}_{\substack{t,\, x_t^w \sim q(x_t \mid x_0^w) \\ x_t^l \sim q(x_t \mid x_0^l)}} \log \sigma \left(\begin{array}{c} \beta T \Big(\mathrm{KL}(q(x_{t-1}^w \mid x_{0,t}^w) \,\|\, \hat{p}_\theta(x_{t-1}^w \mid x_t^w)) - \mathrm{KL}(q(x_{t-1}^w \mid x_{0,t}^w) \,\|\, p_\theta(x_{t-1}^w \mid x_t^w)) \\ -\mathrm{KL}(q(x_{t-1}^l \mid x_{0,t}^l) \,\|\, \hat{p}_\theta(x_{t-1}^l \mid x_t^l)) + \mathrm{KL}(q(x_{t-1}^l \mid x_{0,t}^l) \,\|\, p_\theta(x_{t-1}^l \mid x_t^l)) \Big) \end{array} \right)$$

$$= -\,\mathbb{E}_{t,\epsilon^w,\epsilon^l} \log \sigma \Big(-\beta T\, \omega(\lambda_t) \Big[\|\epsilon^w - \hat{\epsilon}_\theta(x_t^w, t)\|^2 - \|\epsilon^w - \epsilon_\theta(x_t^w, t)\|^2 \\ - \|\epsilon^l - \hat{\epsilon}_\theta(x_t^l, t)\|^2 + \|\epsilon^l - \epsilon_\theta(x_t^l, t)\|^2 \Big] \Big)$$

$x_t^w \sim q(x_t \mid x_0^w)$, $x_t^w \sim q(x_t \mid x_0^l)$, $t \sim [0,1]$, $\epsilon^w, \epsilon^l \sim \mathcal{N}(0, I)$

Diffusion-DPO summary

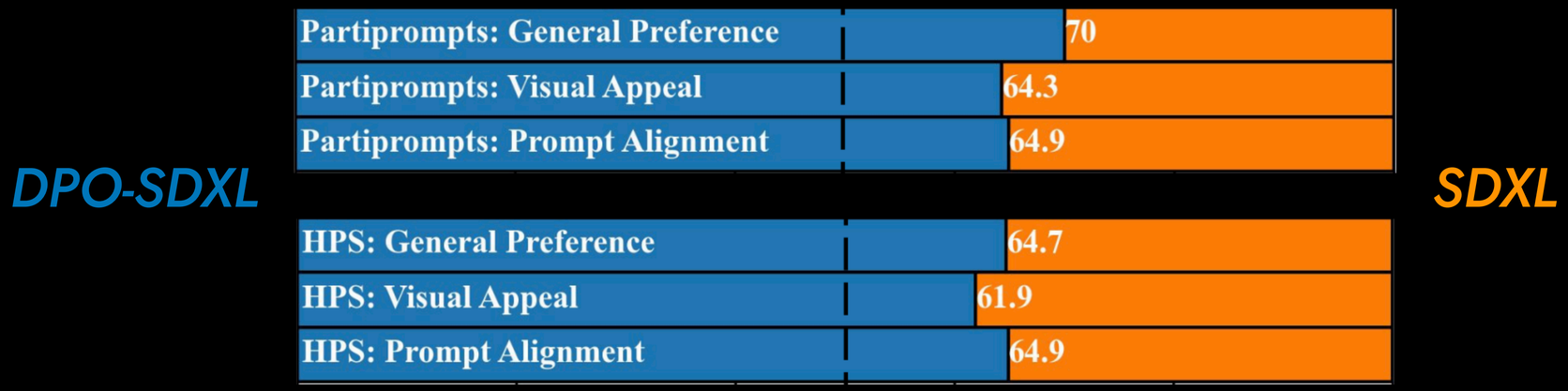
$$L_{\text{DPO-Diffusion}}(\theta) \leq -\mathbb{E}_{t,\epsilon^w,\epsilon^l} \log \sigma\Big(-\beta T\, \omega(\lambda_t) \Big[\|\epsilon^w - \hat{\epsilon}_\theta(x_t^w, t)\|^2 - \|\epsilon^w - \epsilon_\theta(x_t^w, t)\|^2 - \|\epsilon^l - \hat{\epsilon}_\theta(x_t^l, t)\|^2 + \|\epsilon^l - \epsilon_\theta(x_t^l, t)\|^2\Big]\Big)$$

$x_t^w \sim q(x_t \mid x_0^w)$, $x_t^w \sim q(x_t \mid x_0^l)$, $t \sim [0,1]$, $\epsilon^w, \epsilon^l \sim \mathcal{N}(0, I)$

1. Collect a dataset of paired winning and losing samples.
2. Sample one paired sample x_0^w and x_0^l.
3. Select a timestep t.
4. Generate noisy data x_t^w and x_t^l using the forward process q.
5. Compare the model predicted noise $\hat{\epsilon}_\theta$ with the generated noisy data.

Diffusion-DPO evaluation

- Human evaluation to compare the generations under three criteria:
  - Q1. General preference (Which image do you prefer given the prompt?)
  - Q2. Visual appeal (Which image is more visually appealing regardless of the prompt?)
  - Q3. Prompt alignment (Which image better fits the text description?)



Diffusion-DPO examples

Stable Diffusion XL (SDXL)

DPO-SDXL

Application: Diffusion-DPO for safe generation

- As image generation becomes more realistic and widely used, preventing NSFW (Not Safe For Work) content has become increasingly important.
- We will introduce DUO* that utilizes Diffusion-DPO for safe image generation.

Nudity **Violence** **Hate** **Illegality**

Direct Unlearning Optimization for Robust and Safe Text-to-Image Models, Park et al., NeurIPS 2024.

Previous methods for safe image generation

- Most of the previous safety methods depend on malicious text prompts.
  - For example, the model is fine-tuned to generate safe images even when pre-defined malicious keywords (e.g., sexual content, violence) are given.
- Erasing Concepts from Diffusion Models. (Gandikota et al., ICCV 2023)

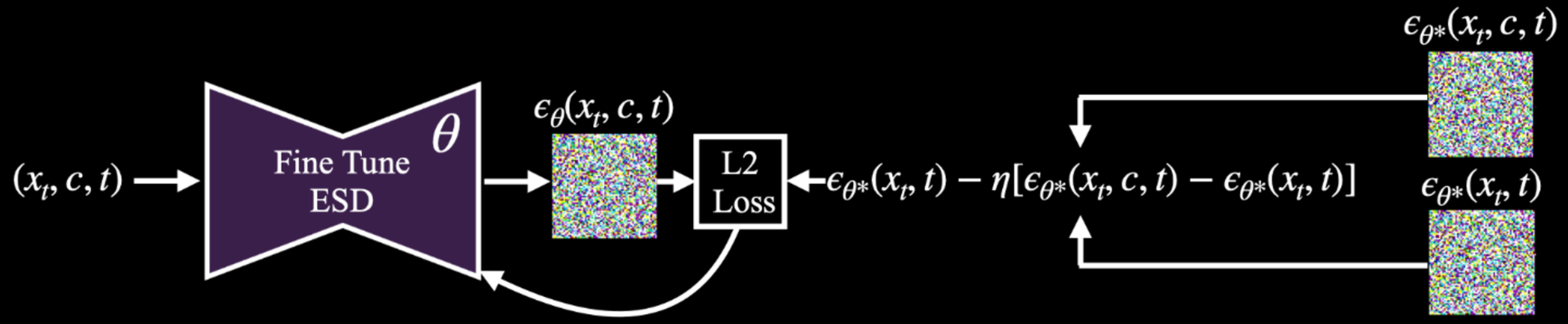
Result of the original model

Erase "Van Gogh" from the model

Result of the edited model

Erasing Concepts from Diffusion Models

- Erasing Concepts from Diffusion Models. (R. Gandikota et al. ICCV 2023)
  - They fine-tune the model to denoise a noised image x_t given a malicious prompt c as it does without the prompt.
  - By subtracting the predicted noises with and without the prompt, we can find the direction related to the malicious prompt and more suppress the model using it.



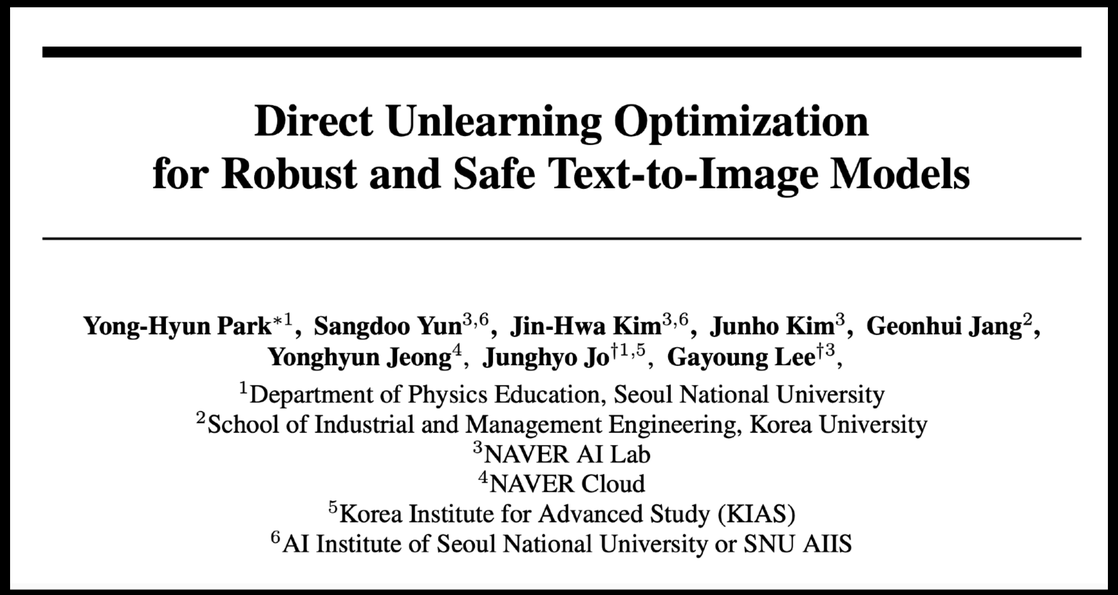
Problem of the previous method

- However, this approach is often prone to adversarial prompt attacks.
  - Adversarial prompt: a seemingly harmless prompt that makes the diffusion model generate unsafe images (e.g., s#i2cn!).
  - This is because the approach fine-tunes the model only for specific prompts, and several studies* have shown that by bypassing those prompts, unsafe images can still be generated.

Ring-A-Bell! How Reliable are Concept Removal Methods For Diffusion Models? Tsai et al., ICLR 2024.

Direct unlearning optimization (DUO)

- To prevent adversarial prompt attacks, the model needs to remove features related to unsafe concepts, regardless of the given prompt.
- How can we remove harmful features from the model without using text prompts?



Direct Unlearning Optimization for Robust and Safe Text-to-Image Models

Yong-Hyun Park[*1], **Sangdoo Yun**[3,6], **Jin-Hwa Kim**[3,6], **Junho Kim**[3], **Geonhui Jang**[2], **Yonghyun Jeong**[4], **Junghyo Jo**[†1,5], **Gayoung Lee**[†3],

[1]Department of Physics Education, Seoul National University
[2]School of Industrial and Management Engineering, Korea University
[3]NAVER AI Lab
[4]NAVER Cloud
[5]Korea Institute for Advanced Study (KIAS)
[6]AI Institute of Seoul National University or SNU AIIS

Direct Unlearning Optimization for Robust and Safe Text-to-Image Models, Park et al., NeurIPS 2024.

Rethinking unlearning as preference optimization

$$L_{\text{DPO-Diffusion}}(\theta) \leq -\mathbb{E}_{t,\epsilon^w,\epsilon^l} \log \sigma\Big(-\beta T\, \omega(\lambda_t) \Big[\big(\|\epsilon^w - \hat{\epsilon}_\theta(x_t^w, t)\|^2 - \|\epsilon^w - \epsilon_\theta(x_t^w, t)\|^2\big) - \big(\|\epsilon^l - \hat{\epsilon}_\theta(x_t^l, t)\|^2 - \|\epsilon^l - \epsilon_\theta(x_t^l, t)\|^2\big)\Big]\Big)$$

$$x_t^w \sim q(x_t \mid x_0^w),\ x_t^w \sim q(x_t \mid x_0^l),\ t \sim [0,1],\ \epsilon^w, \epsilon^l \sim \mathcal{N}(0, I)$$

- x_0^w is a safe image sample and x_0^l is an unsafe image sample in our task.
- Since our method unlearns harmful features without relying on text prompts, it remains robust even when an adversarial prompt is given.

DUO - synthesize paired data

- For preference optimization, we need paired data, (x_0^w, x_0^l), where x_0^w is a set of safe images and x_0^l is a set of unsafe images.
- Unsafe images often contain concepts that are unrelated to the unsafe content.
- How can we selectively remove the unsafe concepts while keeping the rest intact?

Concepts to be removed: Naked, Nudity...

Concepts to be kept: Forest, Human, Girl...

* High-Resolution Image Synthesis with Latent Diffusion Models, Rombach et al., CVPR 2022

** SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations, Meng et al., ICLR 2022.

DUO - synthesize paired data

- For preference optimization, we need paired data, (x_0^w, x_0^l), where x_0^w is a set of safe images and x_0^l is a set of unsafe images.
- Unsafe images often contain concepts that are unrelated to the unsafe content.
- How can we selectively remove the unsafe concepts while keeping the rest intact?

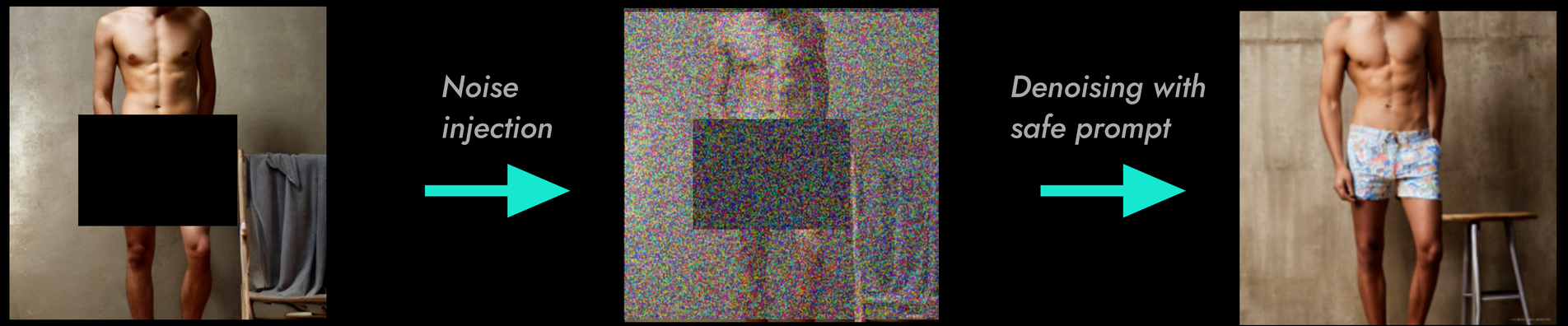
Concepts to be removed: Naked, Nudity…

* High-Resolution Image Synthesis with Latent Diffusion Models, Rombach et al., CVPR 2022

** SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations, Meng et al., ICLR 2022.

DUO - synthesize paired data

- For preference optimization, we need paired data, (x_0^w, x_0^l), where x_0^w is a set of safe images and x_0^l is a set of unsafe images.
- We utilize synthetic data using Stable Diffusion* 1.4 and SDEdit**.
- SDEdit** is an image editing method via noise injection and denoising.

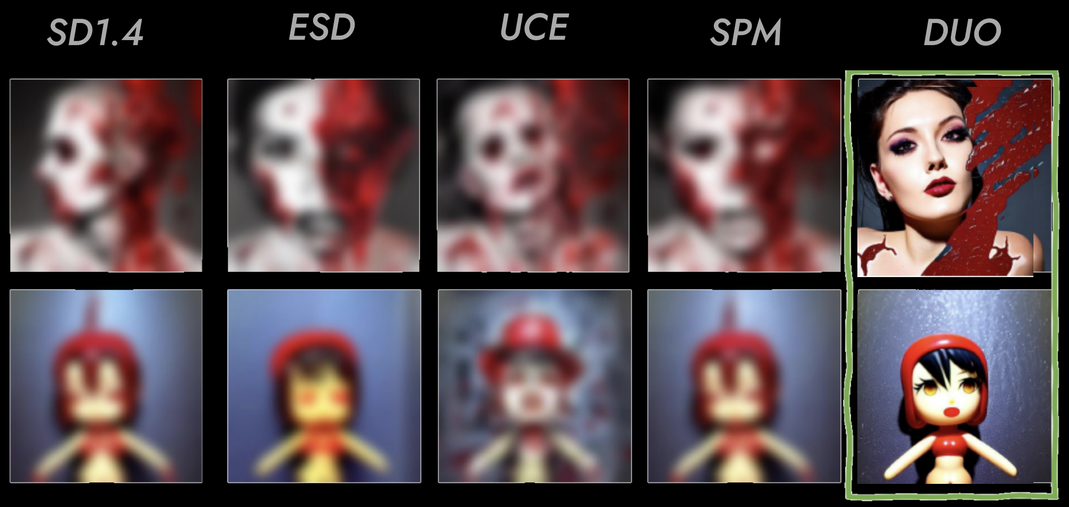


Unsafe image by SD1.4 | *Noised unsafe image* | *Edited safe image*

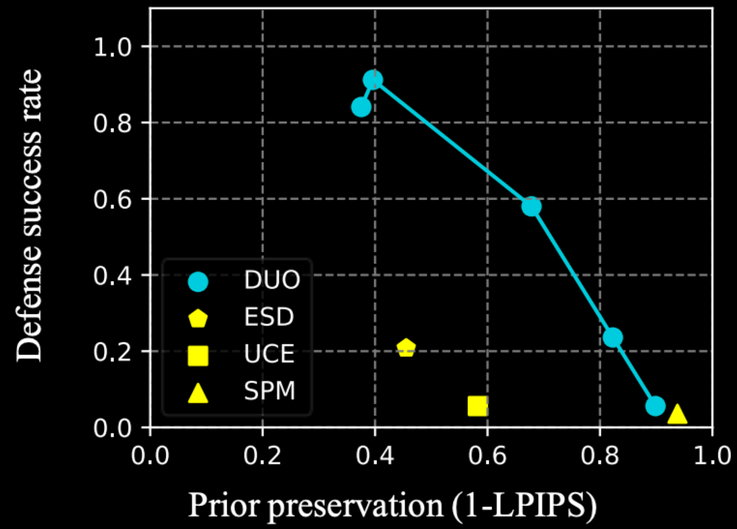
* High-Resolution Image Synthesis with Latent Diffusion Models, Rombach et al., CVPR 2022

** SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations, Meng et al., ICLR 2022.

DUO results



Results using unsafe (violence) prompts after Ring-A-Bell attack



Defense rate and prior preservation with Ring-A-Bell attack

- While previous methods that use prompts for training struggles with adversarial prompt attack (Ring-A-Bell*), our method still prevents unsafe image generation.

Ring-A-Bell! How Reliable are Concept Removal Methods For Diffusion Models? Tsai et al., ICLR 2024

DUO results

SD1.4

SD1.4 + DUO

- Results using prompts from the MS COCO validation after removing nudity.
- For unrelated prompts, the fine-tuned model keeps the original performance.

Summary

- *Preference optimization* refers to training (usually fine-tuning) a model to align with human preferences as specified by a reward model.
- *Direct Preference Optimization (DPO)* removes the need for an explicit reward model by directly optimizing the loss using paired winning and losing samples.
- *Diffusion-DPO* extends DPO to diffusion models by defining a reward function over multi-timestep trajectories and approximating the DPO loss in the diffusion setting.
- *Direct Unlearning Optimization (DUO)* leverages Diffusion-DPO for unlearning by framing the unlearning task as dispreferring unsafe images.

Thank you!

- Any question?