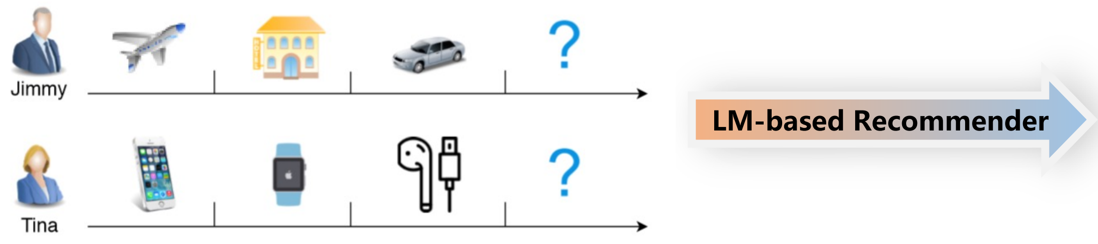
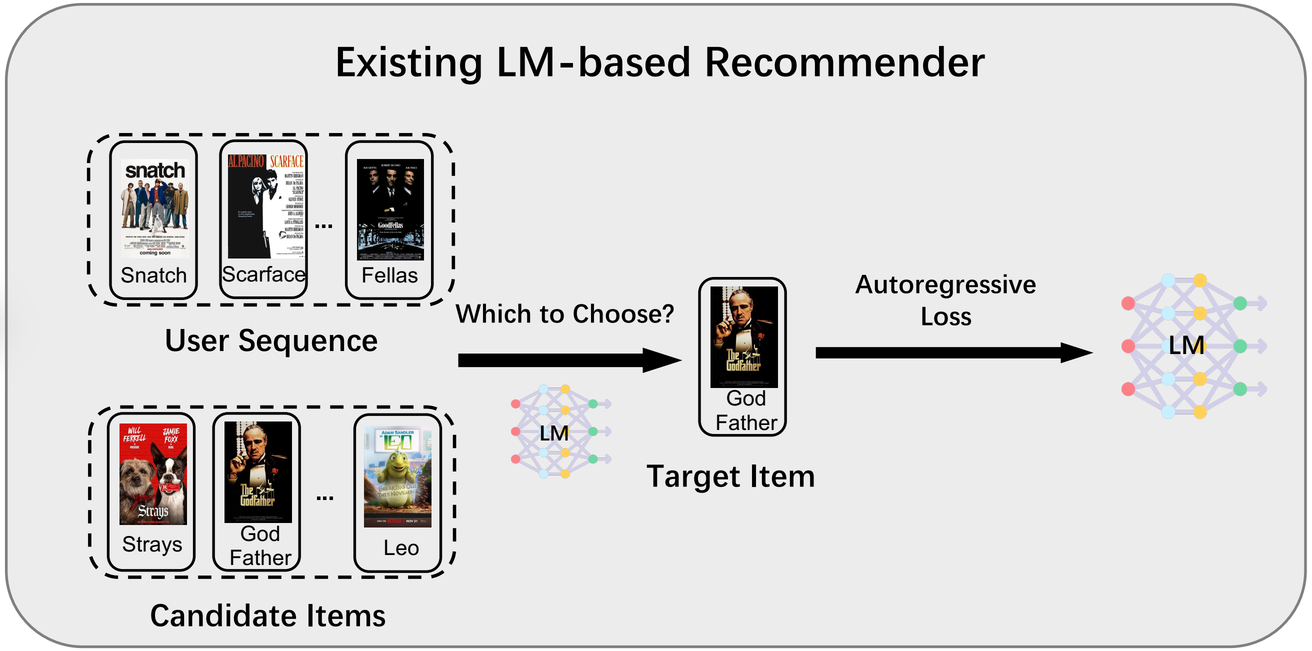
# On Softmax Direct Preference Optimization for Recommendation

- **LM-based recommender have been widely explored.**
  - Extensive world knowledge and strong reasoning ability.
- **Existing LM-based recommenders format recommendation as language generation task.**
  - Convert user sequence into language prompt.
  - Pair sequence with target positive item.
  - Train with language modeling loss.

- **SFT can't fully utilize preference data.**
- **Lack of ranking information.**

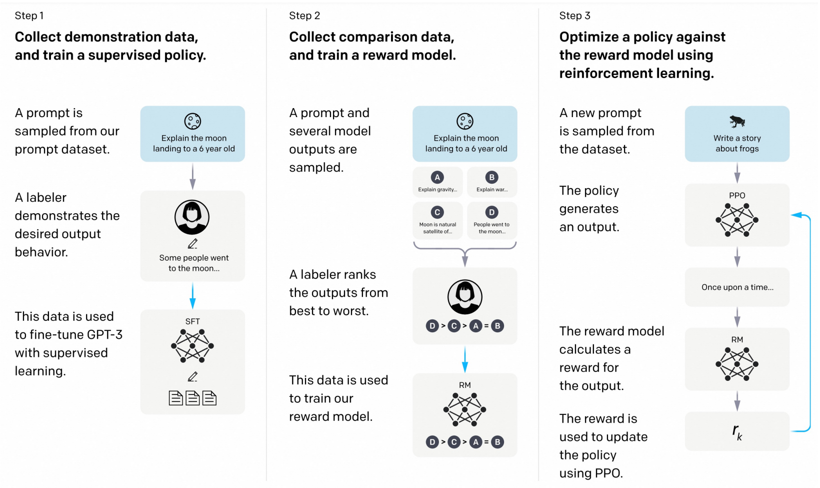



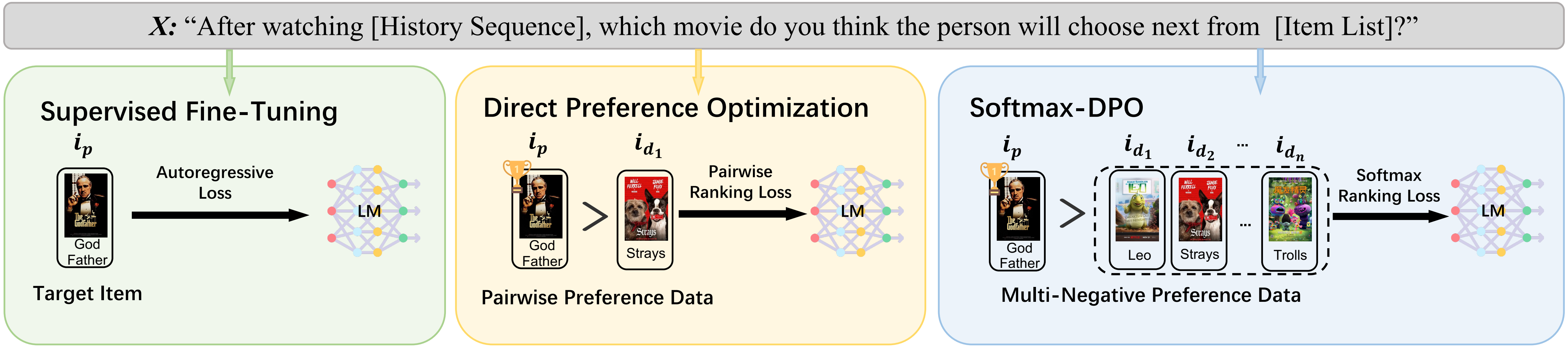



# On Softmax Direct Preference Optimization for Recommendation

## S-DPO

- **Building on SFT, S-DPO:**
  - We make progress on aligning LMs to recommendations by introducing alignment stage, inspired by LM paradigm.
  - Instill ranking information into LM in the light of DPO.
  - Generalize DPO to Softmax-DPO, utilizing multi-negative preference data.



# On Softmax Direct Preference Optimization for Recommendation

## S-DPO

- **Derivation of S-DPO:**
  - DPO is derived from Bradley-Terry model and Plackett-Luce model.

$$p^*(y_1 \succ y_2 \mid x) = \frac{\exp\left(r^*(x, y_1)\right)}{\exp\left(r^*(x, y_1)\right) + \exp\left(r^*(x, y_2)\right)}.$$

$$p^*(\tau | y_1, \ldots, y_K, x) = \prod_{k=1}^{K} \frac{\exp(r^*(x, y_{\tau(k)}))}{\sum_{j=k}^{K} \exp(r^*(x, y_{\tau(j)}))}$$

  - Generalized from Plackett-Luce model, a preference distribution of multi-negative settings can be derived, which takes the following form:

$$p^*(e_p \succ e_d, \forall e_d \in \mathcal{E}_d | x_u) = \frac{\exp(r(x_u, e_p))}{\sum_{j=1}^{K} \exp(r(x_u, e_j))}.$$

  - The loss for multi-negative preference alignment can be derived by replacing Bradley-Terry model with our multi-negative preference distribution:

$$\mathcal{L}_{\mathrm{S-DPO}}(\pi_\theta; \pi_{\mathrm{ref}}) = -\mathbb{E}_{(x_u, e_p, \mathcal{E}_d) \sim \mathcal{D}} \left[ \log \sigma \left( -\log \sum_{e_d \in \mathcal{E}_d} \exp\left( \beta \log \frac{\pi_\theta(e_d | x_u)}{\pi_{\mathrm{ref}}(e_d | x_u)} - \beta \log \frac{\pi_\theta(e_p | x_u)}{\pi_{\mathrm{ref}}(e_p | x_u)} \right) \right) \right].$$

# On Softmax Direct Preference Optimization for Recommendation

## S-DPO

- **Theoretical Analysis:**
  - Connect BPR loss with DPO loss

$$\mathcal{L}_{\mathrm{BPR}} = -\mathbb{E}_{(u,i_p,i_d)}\left[\log\sigma\left(f(u,i_p) - f(u,i_d)\right)\right],$$

$$\mathcal{L}_{\mathrm{DPO}} = -\mathbb{E}_{(x_u,e_p,e_d)}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(e_p|x_u)}{\pi_{\mathrm{ref}}(e_p|x_u)} - \beta\log\frac{\pi_\theta(e_d|x_u)}{\pi_{\mathrm{ref}}(e_d|x_u)}\right)\right],$$

  - Connect sofmtax loss with S-DPO loss

$$\mathcal{L}_{\mathrm{softmax}} = -\mathbb{E}_{(u,i_p,\mathcal{I}_d)}\left[\log\sigma\left(-\log\sum_{i_d\in\mathcal{I}_d}\exp\left(f(u,i_d) - f(u,i_p)\right)\right)\right].$$

$$\mathcal{L}_{\mathrm{S-DPO}}(\pi_\theta;\pi_{\mathrm{ref}}) = -\mathbb{E}_{(x_u,e_p,\mathcal{E}_d)\sim\mathcal{D}}\left[\log\sigma\left(-\log\sum_{e_d\in\mathcal{E}_d}\exp\left(\beta\log\frac{\pi_\theta(e_d|x_u)}{\pi_{\mathrm{ref}}(e_d|x_u)} - \beta\log\frac{\pi_\theta(e_p|x_u)}{\pi_{\mathrm{ref}}(e_p|x_u)}\right)\right)\right].$$

  - Gradient Analysis

$$\nabla_\theta\mathcal{L}_{\mathrm{S-DPO}}(\pi_\theta;\pi_{\mathrm{ref}}) =$$
$$-\beta\mathbb{E}_{(x_u,e_p,\mathcal{E}_d)}\Bigg[\underbrace{\sigma\left(\log\sum_{e_d\in\mathcal{E}_d}\exp(g(e_d,e_p,x_u))\right)}_{\text{higher weight when reward deviates from preference}}\cdot\Bigg[\nabla_\theta\log\pi_\theta(e_p|x_u) - \sum_{e_d\in\mathcal{E}_d}\underbrace{\frac{\nabla_\theta\log\pi_\theta(e_d|x_u)}{\sum_{e'_d\in\mathcal{E}_d}\exp(g(e'_d,e_d,x_u))}}_{\text{higher weight when reward is larger}}\Bigg]\Bigg],$$

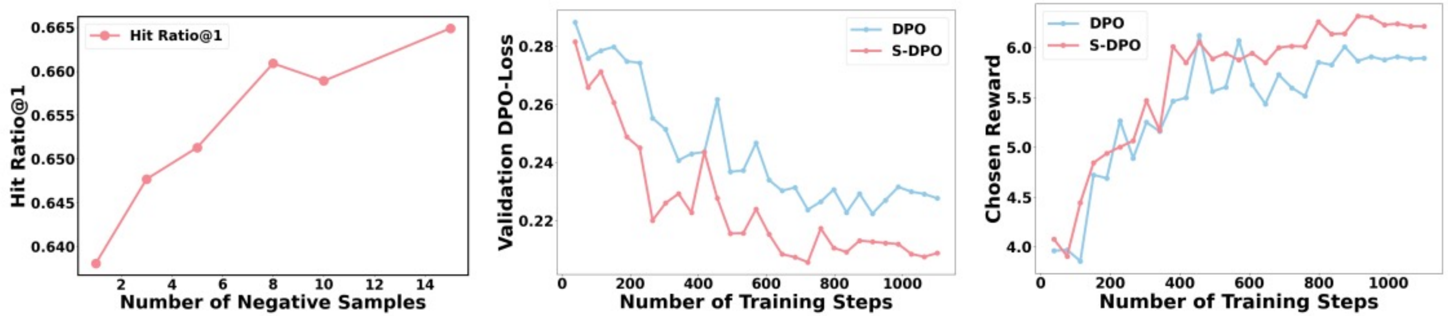
# On Softmax Direct Preference Optimization for Recommendation

- S-DPO achieves significant improvements in sequential recommendations.

Table 1: The performance comparison on three real-world datasets. The improvement achieved by S-DPO is significant ($p$-value << 0.05).

| | | LastFM | | | Goodreads | | | MovieLens | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | HR@1 | ValidRatio | Rel.Ipv | HR@1 | ValidRatio | Rel.Ipv | HR@1 | ValidRatio | Rel.Ipv |
| Traditional | GRU4Rec [45] | 0.3867 | 1.0000 | *70.91%* | 0.2616 | 1.0000 | *153.36%* | 0.3750 | 1.0000 | *40.35%* |
| | Caser [46] | 0.4174 | 1.0000 | *58.34%* | 0.2233 | 1.0000 | *196.82%* | 0.3861 | 1.0000 | *36.31%* |
| | SASRec [47] | 0.3581 | 1.0000 | *84.56%* | 0.2233 | 1.0000 | *196.82%* | 0.3444 | 1.0000 | *52.82%* |
| LM-based | LLaMA2 [31] | 0.0233 | 0.3845 | *2736.48%* | 0.0246 | 0.3443 | *2594.31%* | 0.0421 | 0.4421 | *1150.12%* |
| | ChatRec [51] | 0.3306 | 1.0000 | *99.91%* | 0.3770 | 1.0000 | *75.81%* | 0.2000 | 0.9895 | *163.15%* |
| | MoRec [48] | 0.2877 | 1.0000 | *129.72%* | 0.1652 | 1.0000 | *301.21%* | 0.2822 | 1.0000 | *86.50%* |
| | TALLRec [13] | 0.4983 | 0.9573 | *32.63%* | 0.4180 | 0.9836 | *58.56%* | 0.3895 | 0.9263 | *35.12%* |
| | LLaRA [18] | 0.5292 | 0.9950 | *24.89%* | 0.4508 | 0.9918 | *47.03%* | 0.4737 | 0.9684 | *11.10%* |
| Ours | S-DPO | **0.6609** | 0.9900 | - | **0.6628** | 0.9992 | - | **0.5263** | 0.9895 | - |

- Mining hard negatives brings effective gradients.
- Multi-negatives can provide more reward to preferred items.




(a) Study of performance. (b) Study of validation loss. (c) Study of preferred item reward.

# On Softmax Direct Preference Optimization for Recommendation

- The superiority of S-DPO can be generalized to other LM backbones

Table 4: The performance comparison among three different backbone language models on LastFM and MovieLens.

| | | LLaMA1-7B | | Mistral-7B | | Pythia-2.8B | |
|---|---|---|---|---|---|---|---|
| | | HR@1 | ValidRatio | HR@1 | ValidRatio | HR@1 | ValidRatio |
| LastFM | Vanilla | 0.0465 | 0.5872 | 0.0633 | 0.3648 | 0.0265 | 0.3648 |
| | Language Modeling | 0.5980 | 0.9980 | **0.7828** | 0.9992 | 0.1611 | 0.4281 |
| | DPO | 0.6084 | 0.9976 | 0.7415 | 0.9964 | 0.1896 | 0.4220 |
| | S-DPO (3 negatives) | 0.6285 | 0.9976 | 0.7679 | 0.9972 | 0.1948 | 0.4689 |
| | S-DPO (8 negatives) | **0.6365** | 0.9988 | 0.7820 | 0.9972 | **0.2200** | 0.4685 |
| MovieLens | Vanilla | 0.0316 | 0.5158 | 0.0842 | 0.6737 | 0.0421 | 0.4421 |
| | Language Modeling | 0.3895 | 0.9684 | 0.4211 | 0.9895 | 0.1053 | 0.5684 |
| | DPO | 0.3789 | 0.9684 | 0.4421 | 0.9684 | 0.1271 | 0.8449 |
| | S-DPO (3 negatives) | **0.4526** | 0.9474 | 0.4421 | 0.9895 | 0.1271 | 0.8737 |
| | S-DPO (8 negatives) | **0.4526** | 0.9579 | **0.4947** | 0.9895 | **0.1474** | 0.8737 |

- S-DPO have comparable effectiveness and better efficiency compared with multi-negative DPO variants

Table 2: Effectiveness comparison between DPO with single negative, a variant of DPO with multiple negatives and S-DPO with the same number of negatives (we set $K$ as 3 to get the performance in this table).

| **Datasets** | **LastFM** | | **MovieLens** | | **Goodreads** | | **Complexity** |
|---|---|---|---|---|---|---|---|
| **Measure** | **HitRatio@1** | **ValidRatio** | **HitRatio@1** | **ValidRatio** | **HitRatio@1** | **ValidRatio** | |
| **DPO-1negative** | 0.6342 | 0.9972 | 0.4947 | 0.9684 | 0.6381 | 0.9900 | $\Theta(2C_{\mathcal{M}}S_t)$ |
| **DPO-$K$negative** | 0.6413 | 0.9964 | 0.4947 | 0.9474 | 0.6628 | 0.9900 | $\Theta(2KC_{\mathcal{M}}S_t)$ |
| **S-DPO-$K$negative** | **0.6477** | 0.9980 | **0.5263** | 0.9895 | **0.6661** | 0.9950 | $\Theta((K+1)(C_{\mathcal{M}}+1)S_t)$ |

# *Thanks!*