



Advantage-Guided Distillation for Preference Alignment in Small Language Models

Shiping Gao, Fanqi Wan, Jiajian Guo, Xiaojun Quan, Qifan Wang



Published: 23 Jan 2025, Last Modified: 02 Mar 2025



ICLR 2025 Spotlight



Everyone



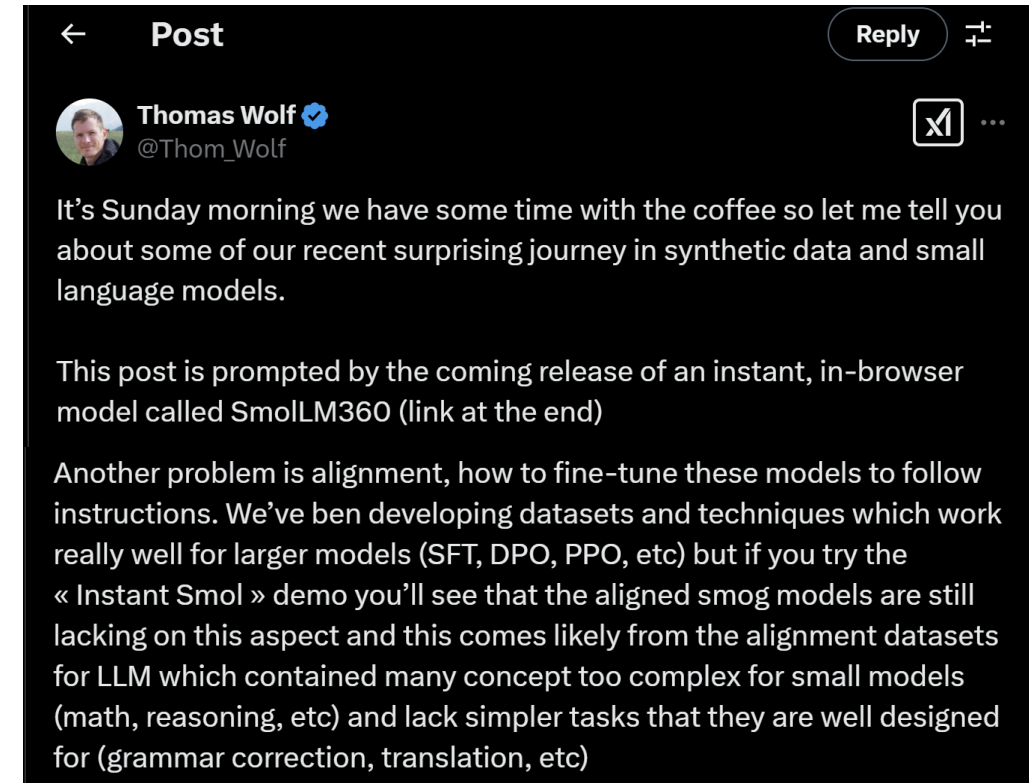
Revisions





Background

- ❑ **End-device needs:** Preference-aligned large language models (LLMs) face significant deployment challenges on end devices.
- ❑ **Alignment Tax:** Small language models (SLMs) often experience performance degradation after preference alignment.
- ❑ **Industrial Drive:** Practical innovations from industry leaders like Hugging Face (e.g., SmolLM360) are driving research momentum toward efficient and deployable small-scale language models.
- ❑ **Our Contributions:** Combine Knowledge Distillation with Preference Alignment, using LLMs to teach SLMs for better performance
- ❑ We introduce the DCKD and ADPA methods, greatly align SLMs' responses with human preference, finally accepted by **ICLR2025 Spotlight**.



Thomas Wolf (CSO of HuggingFace)

Reference:

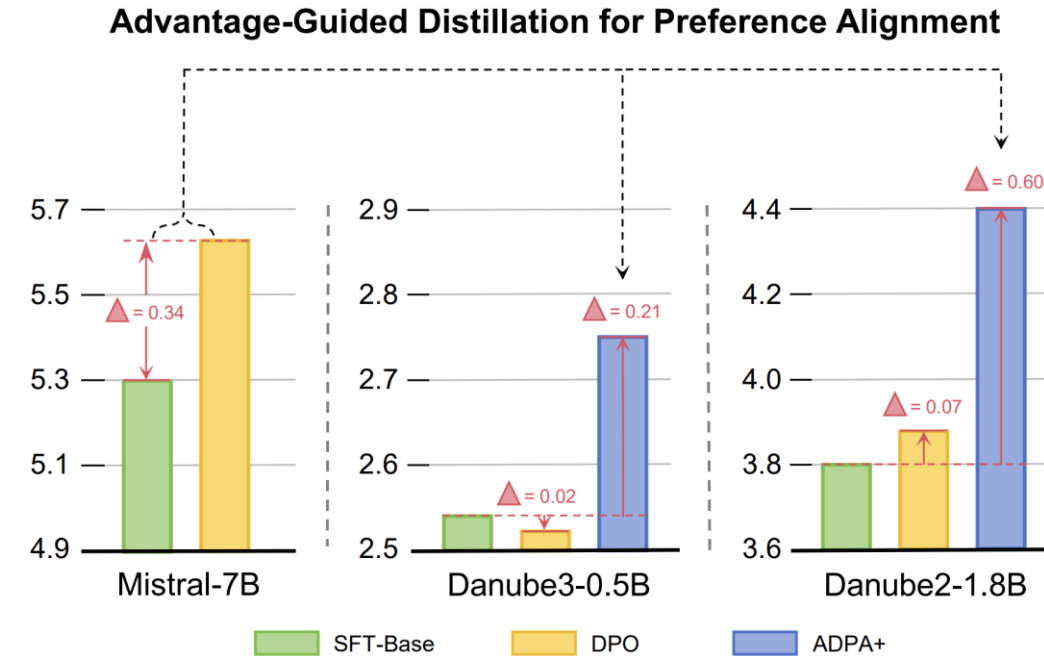
[1] Bai Y, Jones A, Ndousse K, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback[J]. arXiv preprint arXiv:2204.05862, 2022.

[2] Pham T M, Nguyen P T, Yoon S, et al. SlimLM: An Efficient Small Language Model for On-Device Document Assistance[J]. arXiv preprint arXiv:2411.09944, 2024.



Background

- ❑ **End-device needs:** Preference-aligned large language models (LLMs) face significant deployment challenges on end devices.
- ❑ **Alignment Tax:** Small language models (SLMs) often experience performance degradation after preference alignment.
- ❑ **Industrial Drive:** Practical innovations from industry leaders like Hugging Face (e.g., SmolLM360) are driving research momentum toward efficient and deployable small-scale language models.
- ❑ **Our Contributions:** Combine Knowledge Distillation with Preference Alignment, using LLMs to teach SLMs for better performance;
- ❑ We introduce the DCKD and ADPA methods, greatly align SLMs' responses with human preference, finally accepted by **ICLR2025 Spotlight**.



Reference:

[1] Bai Y, Jones A, Ndousse K, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback[J]. arXiv preprint arXiv:2204.05862, 2022.

[2] Pham T M, Nguyen P T, Yoon S, et al. SlimLM: An Efficient Small Language Model for On-Device Document Assistance[J]. arXiv preprint arXiv:2411.09944, 2024.



Dual-Constrained Knowledge Distillation (DCKD)

[stage-1 of our pipeline]

- For y_w is preferred and y_l is dispreferred responses in preference dataset, and π_{dpo} as the DPO-trained teacher, two KL-divergence constraints are defined:

- KLD on preferred response: $\mathcal{L}_{\text{KLD}-w}(\pi_{\text{dpo}}, \pi_{\theta}) = E_{(x, y_w) \sim D} [\sum_{t=1}^{|y_w|} D_{\text{KL}}(\pi_{\text{dpo}}(\cdot | x, y_{w, < t})) || \pi_{\theta}(\cdot | x, y_{w, < t}))]$

- KLD on dispreferred response: $\mathcal{L}_{\text{KLD}-l}(\pi_{\text{dpo}}, \pi_{\theta}) = E_{(x, y_l) \sim D} [\sum_{t=1}^{|y_l|} D_{\text{KL}}(\pi_{\text{dpo}}(\cdot | x, y_{l, < t})) || \pi_{\theta}(\cdot | x, y_{l, < t}))]$

- Including the supervised fine-tuning (SFT) term, the overall objective of DCKD is:

$$\mathcal{L}_{\text{DCKD}} = \mathcal{L}_{\text{SFT}} + \alpha(\mathcal{L}_{\text{KLD}-w} + \mathcal{L}_{\text{KLD}-l})$$

- Leverages preference data to transfer knowledge from an aligned LLM (teacher) to an unaligned SLM (student).
- Enable SLM to capture **both positive signals (y_w) and negative signals (y_l)**;



Advantage-Guided Distillation for Preference Alignment (ADPA)

[stage-2 of our pipeline]

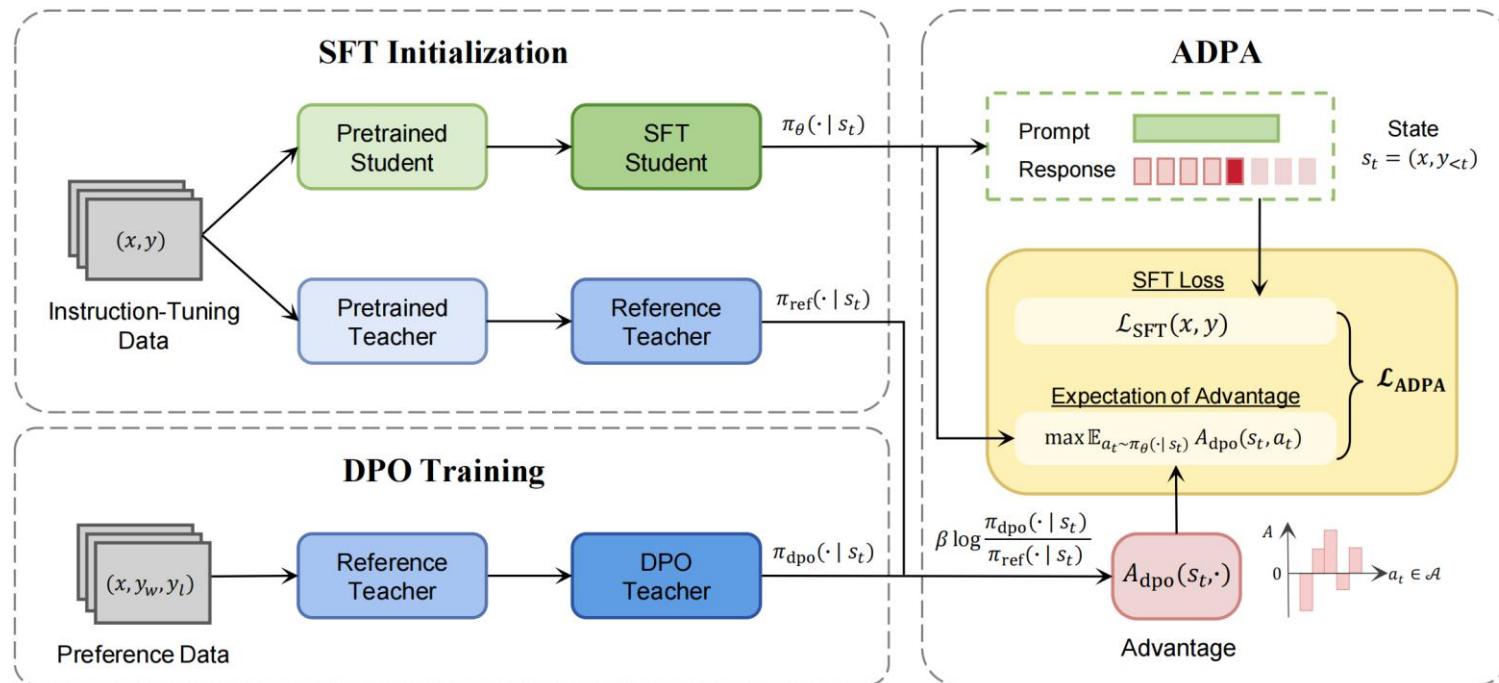
- ❑ The **advantage function** in ADPA is calculated by: $A_{\text{dpo}}(\cdot | s_t) = \beta \cdot \log \pi_{\text{dpo}}(\cdot | s_t) - \beta \cdot \log \pi_{\text{ref}}(\cdot | s_t)$
- ❑ Based on **advantage function** defined above, ADPA's loss is defined in a **Policy-Gradient** Manner:

$$\mathcal{L}_{\text{ADPA}} = E_{(x, y_w) \sim D, \hat{y} \sim \pi_\theta} [\mathcal{L}_{\text{SFT}}(x, y) - \gamma \sum_{t=1}^{|\hat{y}|} \sum_{a_t \in \mathcal{A}} \pi_\theta(a_t | \hat{s}_t) \cdot A_{\text{dpo}}(a_t | \hat{s}_t)]$$

Maximize the Expectation of Advantage

❑ Advantage:

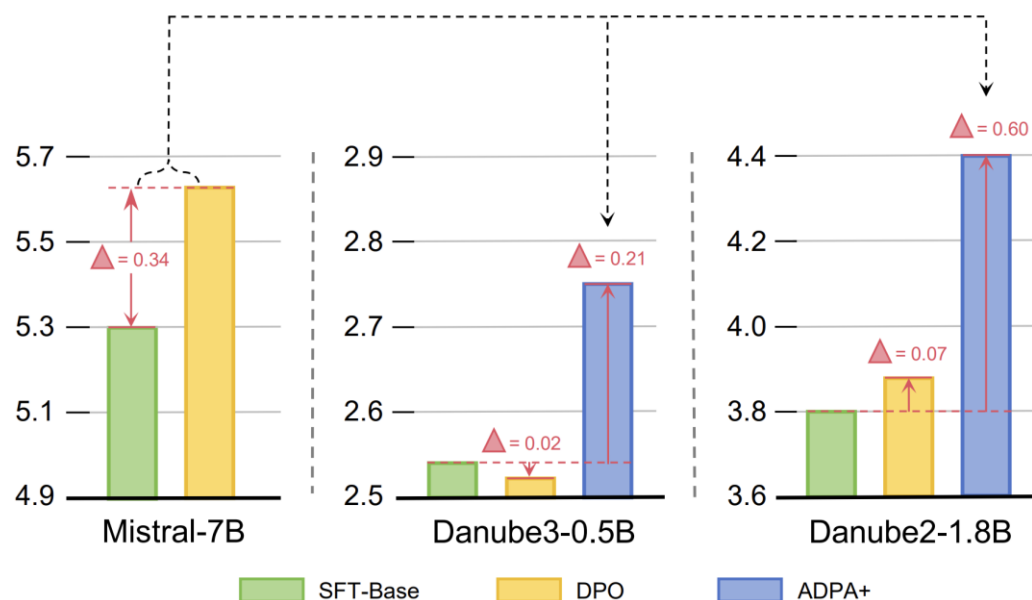
- ❑ Provide **Distribution-Level** reward signals;
- ❑ Efficient use of larger LLM (teacher);
- ❑ Use ADPA to train a DCKD-trained student can further boost its performance.





Experiment

Advantage-Guided Distillation for Preference Alignment



Method	DPO-MIX-7K			HelpSteer2		
	MT-Bench	AlpacaEval WR (%)	OLL	MT-Bench	AlpacaEval WR (%)	OLL
Teacher	6.26	81.7	64.24	6.38	90.3	63.78
Student	3.29	25.1	42.00	3.29	36.8	42.00
SFT	3.34	35.7	42.00	3.13	38.7	41.86
DPO	3.40	33.2	42.70	3.38	39.3	42.47
SimPO	3.37	21.3	42.30	3.47	43.6	42.78
WPO	3.68	39.4	42.67	3.51	48.7	42.85
VanillaKD	3.40	34.1	42.53	3.58	40.2	41.69
SeqKD	3.74	29.7	42.17	3.44	44.4	41.78
ATKD	3.62	32.4	42.28	3.59	42.2	42.42
PLaD	3.42	29.3	42.31	3.36	37.8	42.50
DDPO	3.21	28.7	42.02	3.34	37.3	42.23
DPKD	3.29	28.9	41.87	3.10	36.5	41.74
DCKD	3.50	37.5	42.69	3.44	40.5	41.67
ADPA	3.88	50.0	43.38	3.62	50.0	42.60
ADPA+	4.02	53.8	43.03	3.99	60.9	43.07

Experiment

Dataset: Deita-10K+DPO-MIX-7K & Helpsteer2

Model: Mistral-7B⇒Danube2-1.8B, Danube2-0.5B;

LLaMA3.1-8B⇒LLaMA3.2-1B; LLaMA2-13B⇒LLaMA2-7B;

Benchmark: OpenLLM-Leaderboard, MT-Bench, AlpacaEval (ADPA as reference);

Results: ADPA+ (DCKD plus ADPA) achieves the best performance at most conditions;



Ablation Study

- ❑ **DCKD w/o DPO teacher:** Using Reference teacher to do KD training;
- ❑ **DCKD w/o dispreferred response:** Only do KD training on preferred response;

$$\mathcal{L}_{\text{DCKD}} = \mathcal{L}_{\text{SFT}} + \alpha(\mathcal{L}_{\text{KLD}-w} + \mathcal{L}_{\text{KLD}-l})$$

$$\rightarrow \mathcal{L}_{\text{DCKD}} = \mathcal{L}_{\text{SFT}} + \alpha \mathcal{L}_{\text{KLD}-w}$$

- ❑ **ADPA w/o reference teacher:** Reverse Cross-Entropy KD with DPO teacher;

$$\mathcal{L}_{\text{ADPA}} = E_{(x, y_w) \sim D, \hat{y} \sim \pi_\theta} [\mathcal{L}_{\text{SFT}}(x, y) - \gamma \sum_{t=1}^{|\hat{y}|} \sum_{a_t \in \mathcal{A}} \pi_\theta(a_t | \hat{s}_t) \cdot \beta \cdot \log \pi_{\text{dpo}}(\cdot | s_t) - \beta \cdot \log \pi_{\text{ref}}(\cdot | s_t)]$$

$$\rightarrow \mathcal{L}_{\text{ADPA}} = E_{(x, y_w) \sim D, \hat{y} \sim \pi_\theta} [\mathcal{L}_{\text{SFT}}(x, y) - \gamma \sum_{t=1}^{|\hat{y}|} \sum_{a_t \in \mathcal{A}} \pi_\theta(a_t | \hat{s}_t) \cdot \beta \cdot \log \pi_{\text{dpo}}(\cdot | s_t)]$$

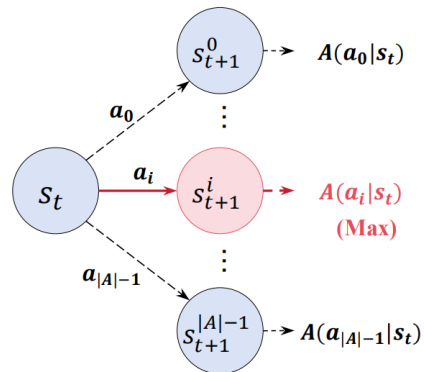
Table 2: Results of ablation for DCKD and ADPA using DPO-MIX-7K dataset. Best performances are shown in **bold**, while second-best are underlined.

Method	Mistral-7B → Danube3-500M		Mistral-7B → Danube2-1.8B		LLaMA-2-13B → LLaMA-2-7B	
	AlpacaEval	WR (%) MT-Bench	AlpacaEval	WR (%) MT-Bench	AlpacaEval	WR (%) MT-Bench
DCKD	50.0	2.67	50.0	4.09	50.0	4.96
- w/o DPO teacher	48.2	2.46	35.6	3.63	39.1	4.60
- w/o dispreferred response	40.3	2.60	39.9	4.04	37.9	4.68
ADPA	50.0	2.56	50.0	4.12	50.0	4.53
- w/o reference teacher	31.6	2.43	36.6	3.78	46.2	4.45

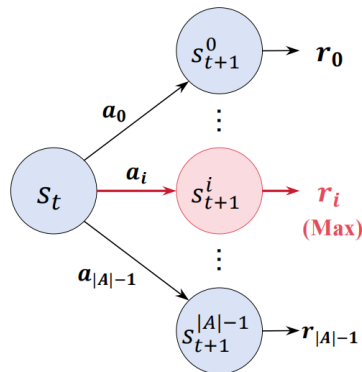


Sample Complexity Analysis

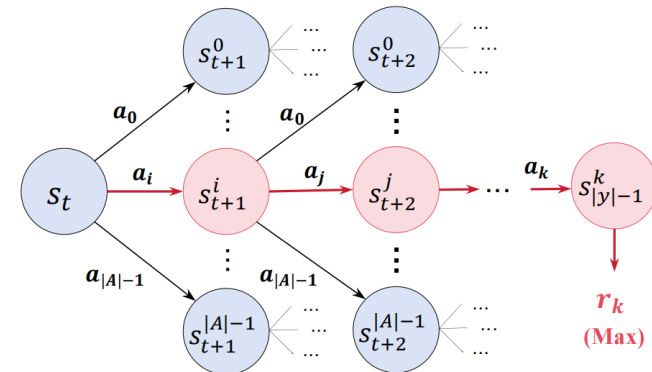
Distribution-Level Advantage




Token-Level Reward



Sequence-Level Reward



 : the next state following the optimal action

□ **Distribution-level Advantage:** $A_{dpo}(\cdot | s_t) = \beta \log \pi_{dpo}(\cdot | s_t) - \beta \log \pi_{ref}(\cdot | s_t)$.

□ Calculate the optimal action through $a^* = \operatorname{argmax}_{a_t \in \mathcal{A}} \beta \log \frac{\pi_{dpo}(a_t | s_t)}{\pi_{ref}(a_t | s_t)}$.

□ Without the need of sampling future states or actions. Sample complexity: $O(1)$.

□ **Token-level Reward:** $r_{\text{token level}}(\{x, y_{<t}\}, y_t) = \beta \log \frac{\pi_{dpo}(a_t | s_t)}{\pi_{ref}(a_t | s_t)}$.

□ Calculating $r(s_t, a_t)$ involves **enumerating all actions** $a_t \in \mathcal{A}$, and computing rewards $r_{\text{token level}}$.

□ Sample complexity: $O(|\mathcal{A}|)$.

□ **Sequence-level Reward:** $r_{\text{seq level}}(x, y) = \beta \sum_{t=1}^{|y|} \log \frac{\pi_{dpo}(y_t | x, y_{<t})}{\pi_{ref}(y_t | x, y_{<t})}$.

□ The model needs to consider **all possible sequences starting from t** to estimate the future rewards.

□ Sample complexity: $O(|\mathcal{A}|^{T-t})$



Sample Complexity Analysis

Table 3: Comparison of ADPA (distribution-level) and PPO-based DPPO with different reward granularities. The sample complexities $O(1)$, $O(|\mathcal{A}|)$, and $O(|\mathcal{A}|^{T-t})$ highlight a theoretical view of how many enumerations or simulations might be needed to find an optimal next action.

Method	Sample Complexity	Reference	AlpacaEval WR (%)
DPPO (sequence-level)	$O(\mathcal{A} ^{T-t})$	ADPA	27.7
DPPO (token-level)	$O(\mathcal{A})$	ADPA	40.0
ADPA (distribution-level)	$O(1)$	ADPA	50.0

□ Lower Sample Complexity

- Works well even offline-RL;
- More stable training;
- Better performance;

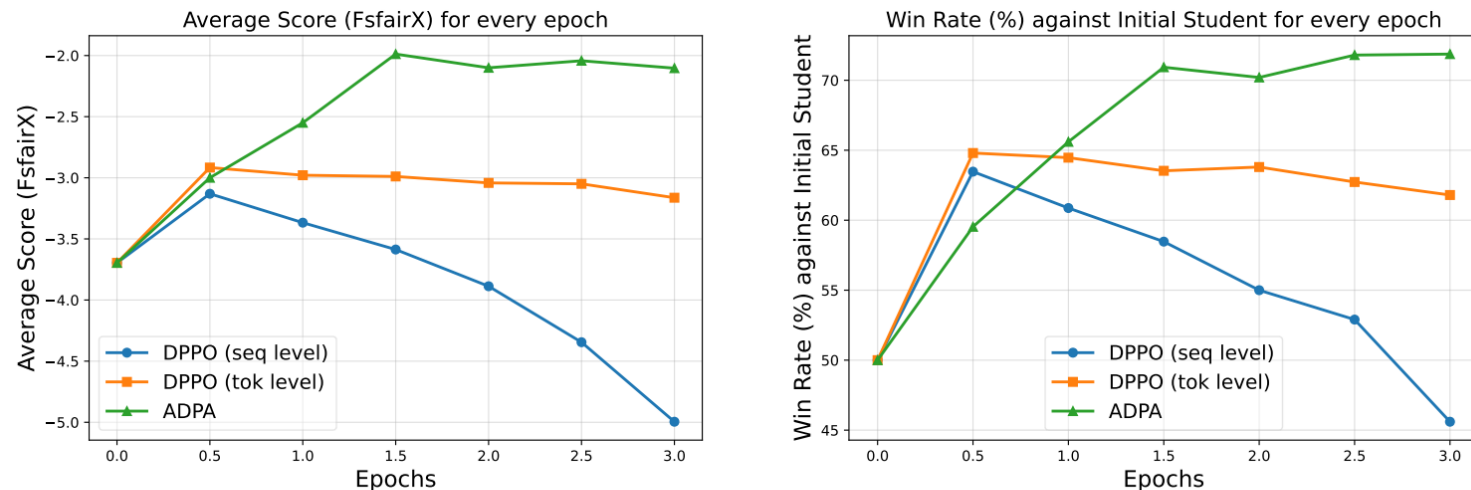


Figure 4: Comparison between ADPA and PPO-based methods on the validation set. The x-axis denotes the training epochs, and the y-axis indicates either the average scores (**left**) or the win rates (**right**) of responses generated by checkpoints during training, as evaluated using FsfairX.



Different KD based on Q & Advantage function

Here, we also compare ADPA(maximizing the advantage) with other kind of KD objectives

In traditional policy KD, the Q function or advantage function is often distilled by softmax or argmax operation combined with KL divergence or cross entropy loss, such as the following Q-argmax and Q-softmax KD.

$$\mathcal{L}_{\text{Q-argmax}} = \mathbb{E}_{(x,y,\hat{y}) \sim \hat{\mathcal{D}}} \left[\mathcal{L}_{\text{SFT}}(x, y) + \frac{\gamma}{|\hat{y}|} \sum_{t=1}^{|\hat{y}|} \text{CE} \left(\mathbf{1} \left\{ \arg \max_{a_t \in \mathcal{A}} (A_{\text{dpo}}(s_t, a_t)) \right\}, \pi_{\theta}(\cdot | s_t) \right) \right]$$

$$\mathcal{L}_{\text{Q-softmax}} = \mathbb{E}_{(x,y,\hat{y}) \sim \hat{\mathcal{D}}} \left[\mathcal{L}_{\text{SFT}}(x, y) + \frac{\gamma}{|\hat{y}|} \sum_{t=1}^{|\hat{y}|} D_{\text{KL}} (\text{softmax}(A_{\text{dpo}}(s_t, \cdot)) || \pi_{\theta}(\cdot | s_t)) \right]$$

Reference:

[1] Rusu, A. A., Colmenarejo, S. G., Gulcehre, C., Desjardins, G., Kirkpatrick, J., Pascanu, R., ... & Hadsell, R. (2015). Policy distillation. *arXiv preprint arXiv:1511.06295*.



Different KD based on Q & Advantage function

Here, we also compare ADPA(maximizing the advantage) with other kind of KD objectives

$$\mathcal{L}_{\text{Q-argmax}} = \mathbb{E}_{(x,y,\hat{y}) \sim \hat{\mathcal{D}}} \left[\mathcal{L}_{\text{SFT}}(x, y) + \frac{\gamma}{|\hat{y}|} \sum_{t=1}^{|\hat{y}|} \text{CE} \left(\mathbf{1} \left\{ \arg \max_{a_t \in \mathcal{A}} (A_{\text{dpo}}(s_t, a_t)) \right\}, \pi_{\theta}(\cdot | s_t) \right) \right]$$

$$\mathcal{L}_{\text{Q-softmax}} = \mathbb{E}_{(x,y,\hat{y}) \sim \hat{\mathcal{D}}} \left[\mathcal{L}_{\text{SFT}}(x, y) + \frac{\gamma}{|\hat{y}|} \sum_{t=1}^{|\hat{y}|} D_{\text{KL}} (\text{softmax}(A_{\text{dpo}}(s_t, \cdot)) || \pi_{\theta}(\cdot | s_t)) \right]$$

Method	Reference	WR (%)
Q-argmax KD	ADPA	41.8
Q-softmax KD	ADPA	28.2
ADPA	ADPA	50.0

- Taking ADPA as the baseline, **Q-argmax KD** has a win rate of **41.8%** and **Q-softmax KD** has a win rate of **28.2%**.
- In the distillation process, **retaining the original distribution characteristics of the advantage function** may be more critical to performance improvement.



Thanks for listening

Feel free to ask me questions about this paper.
<https://iclr.cc/media/iclr-2024/Slides/18013.pdf>

My Email: rungao2001@outlook.com