Towards Understanding Valuable Preference Data for Large Language Model Alignment

Zizhuo Zhang, Qizhou Wang, Shanshan Ye, Jianing Zhu,
Jiangchao Yao, Bo Han, Masashi Sugiyama

TMLR Group, Hong Kong Baptist University
University of Technology Sydney
CMIC, Shanghai Jiao Tong University
The University of Tokyo
RIKEN Center for Advanced Intelligence Project

Paper

Code

Zizhuo Zhang
cszzzhang@comp.hkbu.edu.hk

Outline

Background
LLM Preference Alignment
RLHF

Preliminary
Existing Alignment Methods
Influence Function (IF)

Motivation
Preference Data Quality
Influence Function (IF)-based Analysis

Methodology
LossDiff-IRM Data Selection

Experiments
Better Performance using
Less Data

Background: LLM Preference Alignment

Large Language Model (LLM) preference alignment is the process of training LLMs to generate outputs that better match human preferences and values.

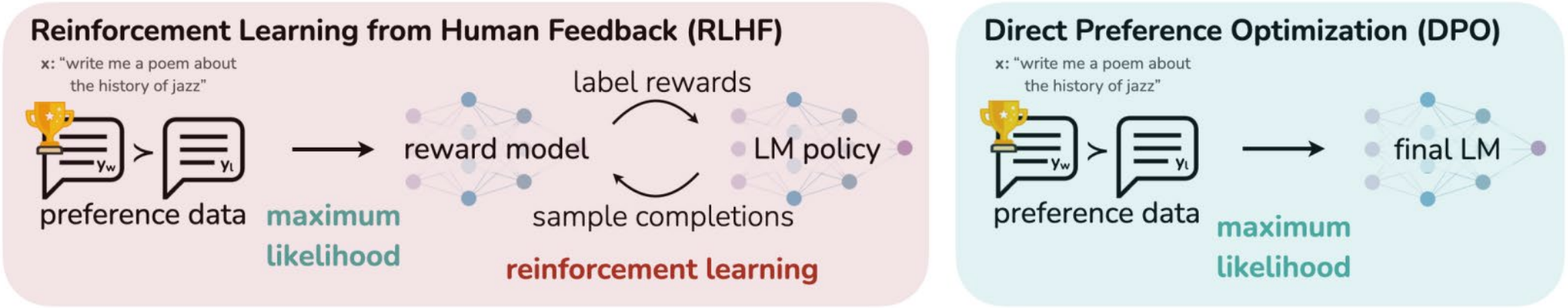


Figure refers to Rafailov et al.

- **Pairwise preference data** $\mathcal{D} = \{\boldsymbol{d} = (\boldsymbol{x}, \boldsymbol{y_w}, \boldsymbol{y_l})\}$, where $y_w \succ y_l$ denotes that the human prefers the chosen response y_w rather than the rejected one y_l for the prompt x.
- **Traditional RLHF:** (1) train a reward model $r_\phi(\cdot)$ using preference data; (2) RL with reward from the reward model:

$$\max_\theta \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}[r_\phi(x,y)] - \beta\mathbb{D}_{\mathrm{KL}}[\pi_\theta(y|x)||\pi_{\mathrm{ref}}(y|x)]$$

- **DPO and its improved versions:** directly align the model on preference data:

$$\mathcal{L}_{\mathrm{DPO}}(\theta;\mathcal{D}) = -\mathbb{E}_{(x,y_w,y_l)\in\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w|x)}{\pi_{\mathrm{ref}}(y_w|x)} - \beta\log\frac{\pi_\theta(y_l|x)}{\pi_{\mathrm{ref}}(y_l|x)}\right)\right]$$

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

Preliminary: DPO series & Influence Function (IF)

Traditional RLHF follows a two-stage training pipeline, making it relatively complex, whereas the **DPO series directly optimizes the model from preference data,** enabling more straightforward analysis of the impact of preference data:

- **DPO [1]:** $\mathcal{L}_{\mathrm{DPO}}(\theta;\mathcal{D}) = -\mathbb{E}_{(x,y_w,y_l)\in\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w|x)}{\pi_{\mathrm{ref}}(y_w|x)} - \beta\log\frac{\pi_\theta(y_l|x)}{\pi_{\mathrm{ref}}(y_l|x)}\right)\right]$
- **SLiC [2]:** $\mathcal{L}_{\mathrm{SLiC}}(\theta;\mathcal{D}) = \mathbb{E}_{(x,y_w,y_l)\in\mathcal{D}}\left[\max\left(0,1-\left(\beta\log\frac{\pi_\theta(y_w|x)}{\pi_{\mathrm{ref}}(y_w|x)} - \beta\log\frac{\pi_\theta(y_l|x)}{\pi_{\mathrm{ref}}(y_l|x)}\right)\right)\right]$
- **SimPO [3]:** $\mathcal{L}_{\mathrm{SimPO}}(\theta;\mathcal{D}) = -\mathbb{E}_{(x,y_w,y_l)\in\mathcal{D}}\left[\log\sigma\left(\frac{\beta}{|y_w|}\log\pi_\theta(y_w|x) - \frac{\beta}{|y_l|}\log\pi_\theta(y_l|x) - \gamma\right)\right]$

Influence Function (IF) is a classical analytic tool that **estimates how a training sample affects validation performance** by measuring the alignment between its gradient and the validation gradient:

$$\mathrm{IF}(d;\pi_\theta;\mathcal{D}_{\mathrm{val}}) = \left(\frac{1}{|\mathcal{D}_{\mathrm{val}}|}\sum_{i}^{|\mathcal{D}_{\mathrm{val}}|}\nabla_\theta\ell\left(\theta;d_{\mathrm{val}}^{(i)}\right)\right)^{\mathrm{T}}\nabla_\theta\ell(\theta;d)$$

where d is a training sample from $\mathcal{D}$, and **a higher IF means the training gradient better aligns with the validation gradient**, suggesting the data is more beneficial for generalization.

[1] *Rafailov et al. Direct Preference Optimization: Your Language Model is Secretly a Reward Model*
[2] *Liu et al. Statistical rejection sampling improves preference optimization.*
[3] *Meng et al. SimPO: Simple Preference Optimization with a Reference-Free Reward*

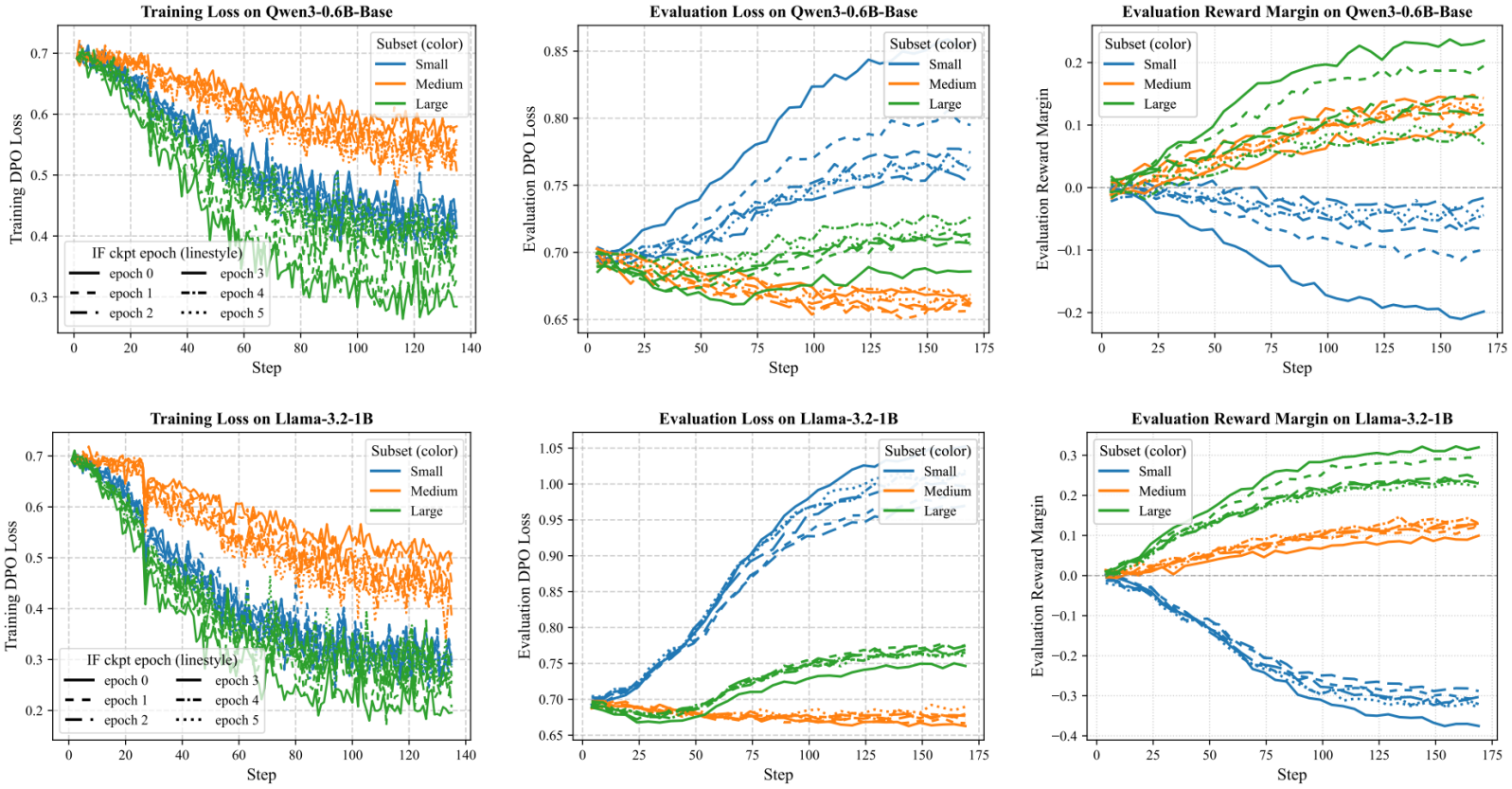
Motivation: IF for Analyzing Preference Data Quality

- Existing studies **treat preference data quality as an intrinsic property**, identifying and selecting valuable preference pairs using external reward models or off-the-shelf LLMs.
- **Our view: preference data quality is model-dependent**, where a preference pair that benefits one model alignment may harm another.
- **IF under DPO for a given preference pair $d = (x, y_w, y_l)$:**

$$\mathrm{IF}_{\mathrm{DPO}}(d; \pi_\theta; \mathcal{D}_{\mathrm{val}}) := \beta(1 - \sigma(\Delta_\theta)) \left\langle \underbrace{\frac{\beta}{|\mathcal{D}_{\mathrm{val}}|} \sum_i \left(1 - \sigma(\Delta_\theta^{(i)})\right) \left(g_w^{(i)} - g_l^{(i)}\right)}_{\text{preference generalization direction w.r.t. validation set}}, \underbrace{g_w - g_l}_{\text{current preference pair direction}} \right\rangle,$$

- where $g_* = \nabla_\theta \log \pi_\theta(y_*|x)$ denotes the gradient of the log-likelihood for response $y_* \in \{y_w, y_l\}$, and $\Delta_\theta = \beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\mathrm{ref}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\mathrm{ref}}(y_l|x)}$ is the reward difference.
- **DPO IF focuses on the gradient difference consistency**, i.e., $g_w - g_l$, with the validation preference set, serving as a proxy to assess quality of a preference pair d.

Motivation: IF for Analyzing Preference Data Quality



- Medium-IF preference pairs are valuable.
- **Truncated Influence Function (TIF):**

$$\mathrm{TIF}(d;\pi_\theta;\mathcal{D}_{\mathrm{val}}) = \mathbb{I}\left[\delta_{\mathrm{small}} < \mathrm{IF}(d;\pi_\theta;\mathcal{D}_{\mathrm{val}}) < \delta_{\mathrm{large}}\right]$$

- δ_{small} and δ_{large} denote threshold percentiles that specify the boundaries of IF value.

- **Small-IF data:** Training loss decreases as expected, but evaluation loss increases while the evaluation reward margin falls below zero. This indicates that small-IF pairs are largely uninformative and of low quality.
- **Large-IF data:** Evaluation loss first decreases then rises, while reward margin keeps increasing, indicating overfitting: large-IF data amplifies a few pairs but harms many others.
- **Medium-IF data:** As training loss decreases, evaluation loss drops and reward margin increases, **showing that medium-IF pairs provide stable, high-quality signals for better preference generalization**.

Methodology: LossDiff-IRM Data Selection

Computing exact IF value requires gradients on both the training and validation sets, which becomes prohibitive in the large-scale models and datasets.

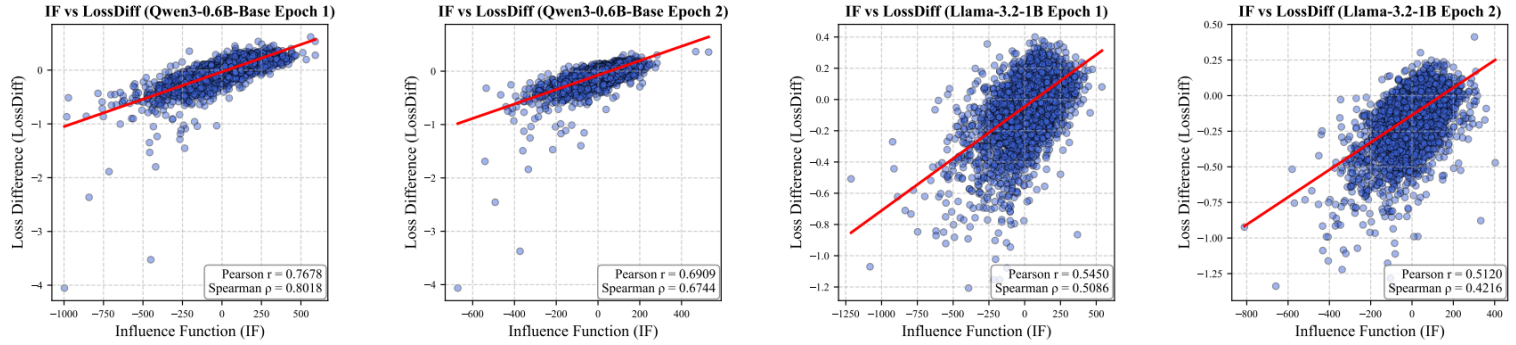
	Computational Time			Throughput Rate (pair/sec)	
IF Computation	Val Gradient	IF	Total	Val Gradient	IF
Qwen3-0.6B-Base	1 h 17 m 31 s	1 h 59 m 45 s	3 h 17 m 16 s	1.55	1.44
Llama-3.2-1B	3 h 55 m 32 s	6 h 02 m 37 s	9 h 58 m 09 s	4.71	4.35
LossDiff-IRM Computation	Training Forward	Val Forward	Total	Training Forward	Val Forward
Qwen3-0.6B-Base	1 min 5 s	59 s	2 m 4 s	76.92	84.74
Llama-3.2-1B	2 m 32 s	2 m 27 s	4 m 59 s	32.86	33.80

We need to find more efficient proxy to approximate the IF value for LLMs.

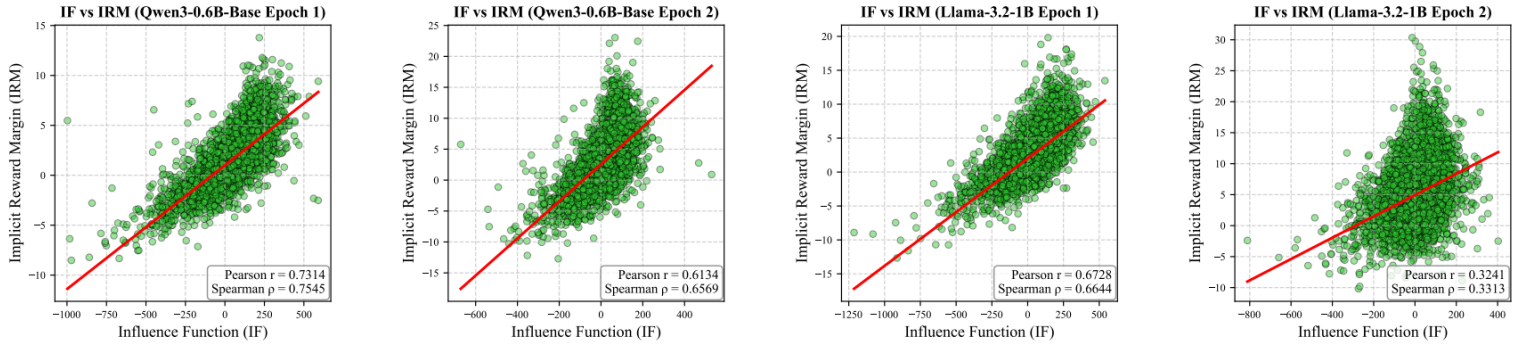
Methodology: LossDiff-IRM Data Selection

Two approximation proxies that are positively correlated with IF:

Loss Difference (LossDiff)



Implicit Reward Margin (IRM)



- **Loss Difference (LossDiff):**

$$\mathrm{LossDiff}\left(d;\pi_\theta;\pi_{\theta_{\mathrm{val}}}\right)=\ell(\theta;d)-\ell(\theta_{\mathrm{val}};d)$$

- where $\pi_{\theta_{\mathrm{val}}}$ is the auxiliary model aligned on validation set.
- **Implicit Reward Margin (IRM):**

$$\mathrm{IRM}_\theta(d)=\beta\log\frac{\pi_\theta(y_w|x)}{\pi_{\mathrm{ref}}(y_w|x)}-\beta\log\frac{\pi_\theta(y_l|x)}{\pi_{\mathrm{ref}}(y_l|x)}$$

Methodology: LossDiff-IRM Data Selection

- **LossDiff-IRM selects a preference pair d if and only if:**

$$\text{LossDiff–IRM}(d;\pi_\theta;\mathcal{D}_{\text{val}}) = \mathbb{I}[\xi_{\text{small}} < \text{LossDiff}(d;\pi_\theta;\mathcal{D}_{\text{val}}) < \xi_{\text{large}}] \\ \wedge\ \mathbb{I}[\tau_{\text{small}} < \text{IRM}(d;\pi_\theta) < \tau_{\text{large}}]\,.$$

- where LossDiff-IRM $\in \{0,1\}$, $(\xi_{\text{small}}, \xi_{\text{large}})$ and $(\tau_{\text{small}}, \tau_{\text{large}})$ are percentile thresholds that define the medium ranges for LossDiff and IRM, respectively.
- **LossDiff-IRM focuses on selecting preference pairs whose LossDiff and IRM values both lie within the medium range.**

Overlap Coefficient	**LossDiff vs. IF**		**IRM vs. IF**		**LossDiff-IRM vs. IF**	
Models	Epoch 1 ckpt	Epoch 2 ckpt	Epoch 1 ckpt	Epoch 2 ckpt	Epoch 1 ckpt	Epoch 2 ckpt
Qwen3-0.6B-Base	0.6953	0.6639	0.6883	0.6470	**0.7820**	**0.7257**
Llama-3.2-1B	0.6687	0.6582	0.6969	0.6025	**0.7657**	**0.6963**

Observation: the combined selector LossDiff-IRM achieves a higher Overlap Coefficient with the exact TIF-selected set than using LossDiff or IRM alone across Qwen3-0.6B-Base and Llama-3.2-1B.

Overall Performance

Methods	Dataset Ratio	UltraFeedback		AlpacaEval		Vicuna-Bench		Arena-Hard	
		Single ↑	WinRate ↑	Single ↑	WinRate ↑	Single ↑	WinRate ↑	Single ↑	WinRate ↑
Llama-3.1-8B (DPO)									
SFT		3.60	-	3.53	-	3.98	-	2.63	-
Full Data	100%	5.77	77.61	5.87	78.41	6.04	73.75	4.68	81.39
Random	64%	5.52	74.83	5.59	75.93	5.46	68.13	4.64	81.27
GPT4	64%	6.04	80.57	6.21	81.09	6.86	80.31	4.96	84.30
Reward Model	64%	6.24	82.68	6.38	83.76	6.45	76.88	5.13	86.19
LossDiff-IRM	64%	6.54	83.97	6.84	87.08	7.06	86.88	5.59	88.40
Pythia-2.8B (DPO)									
SFT		3.94	-	4.35	-	4.66	-	2.74	-
Full Data	100%	4.60	70.53	4.95	67.05	5.35	67.50	2.97	60.71
Random	64%	4.54	68.27	4.79	64.03	5.15	68.13	3.00	63.25
GPT4	64%	4.71	72.02	4.96	67.10	5.39	73.75	3.08	64.15
Reward Model	64%	4.68	75.73	5.09	70.91	5.60	75.95	3.03	63.03
LossDiff-IRM	64%	4.90	79.62	5.30	76.03	5.74	82.50	3.26	71.64
Pythia-410M (DPO)									
SFT		2.56	-	2.47	-	3.15	-	1.91	-
Full Data	100%	2.81	75.25	2.77	73.51	3.10	69.37	2.06	59.47
Random	56%	2.94	76.03	2.92	76.62	3.58	70.63	2.06	57.08
GPT4	56%	2.95	76.78	2.95	79.81	3.49	79.37	2.15	61.44
Reward Model	56%	2.96	81.48	3.02	80.64	3.67	74.38	2.16	60.59
LossDiff-IRM	56%	3.30	86.14	3.30	85.16	3.80	85.62	2.38	69.63
Qwen3-8B-Base (SLiC)									
SFT		6.97	-	6.88	-	7.94	-	6.64	-
Full Data	100%	7.55	59.54	7.61	59.71	8.05	54.69	7.21	59.61
Random	64%	7.57	57.76	7.63	62.05	7.97	47.94	7.25	59.18
GPT4	64%	7.64	59.91	7.89	63.62	8.21	58.37	7.33	59.47
Reward Model	64%	7.74	62.47	8.09	66.57	8.50	63.44	7.38	62.27
LossDiff-IRM	64%	7.87	64.40	8.11	67.58	8.44	61.12	7.61	62.20
Pythia-1.4B (SLiC)									
SFT		3.50	-	3.65	-	4.20	-	2.37	-
Full Data	100%	3.66	63.58	3.98	63.68	4.67	60.00	2.65	58.95
Random	52%	3.81	66.18	3.96	63.99	4.34	56.25	2.66	60.26
GPT4	52%	3.84	69.24	4.12	65.55	4.69	63.75	2.68	59.80
Reward Model	52%	3.82	66.57	4.04	69.20	4.67	65.62	2.67	61.95
LossDiff-IRM	52%	4.14	74.25	4.39	72.69	4.88	71.25	2.85	67.76

Methods	Dataset Ratio	UltraFeedback		AlpacaEval		Vicuna-Bench		Arena-Hard	
		Single ↑	WinRate ↑	Single ↑	WinRate ↑	Single ↑	WinRate ↑	Single ↑	WinRate ↑
Qwen3-8B-Base (DPO)									
SFT		6.97	-	6.88	-	7.94	-	6.64	-
Full Data	100%	7.64	61.41	7.92	63.85	8.21	62.14	7.58	59.61
Random	64%	7.71	61.47	7.94	64.12	8.26	58.93	7.57	62.07
GPT4	64%	7.69	62.19	8.01	63.81	8.28	52.19	7.62	61.53
Reward Model	64%	7.81	64.19	8.24	69.35	8.56	66.25	7.61	64.78
LossDiff-IRM	64%	8.05	67.32	8.36	71.52	8.72	67.19	7.83	68.63
Pythia-1.4B (DPO)									
SFT		3.50	-	3.65	-	4.20	-	2.37	-
Full Data	100%	3.70	65.88	3.99	66.17	4.71	64.38	2.65	60.24
Random	52%	3.78	67.43	4.05	68.16	4.56	61.25	2.60	61.64
GPT4	52%	3.96	70.75	4.28	70.96	4.89	68.75	2.83	64.20
Reward Model	52%	3.83	70.80	4.18	70.83	4.84	68.75	2.71	64.76
LossDiff-IRM	52%	4.23	78.49	4.49	76.43	5.28	76.88	2.96	71.72
Llama-3.1-8B (SLiC)									
SFT		3.60	-	3.53	-	3.98	-	2.63	-
Full Data	100%	5.09	70.72	5.13	72.13	5.40	71.88	3.98	73.75
Random	64%	4.94	69.52	4.89	67.05	5.26	67.50	3.95	70.35
GPT4	64%	5.48	75.61	5.40	72.89	6.05	67.81	4.28	75.27
Reward Model	64%	5.34	73.64	5.54	75.03	5.55	68.75	4.50	78.32
LossDiff-IRM	64%	5.94	79.51	5.84	78.84	5.85	76.56	4.65	83.12
Pythia-2.8B (SLiC)									
SFT		3.94	-	4.35	-	4.66	-	2.74	-
Full Data	100%	4.36	67.46	4.48	61.66	4.90	65.00	2.93	59.04
Random	64%	4.31	63.00	4.56	59.73	5.16	65.62	2.98	57.58
GPT4	64%	4.50	69.09	4.73	63.00	5.12	65.62	2.89	58.38
Reward Model	64%	4.43	70.03	4.76	64.92	5.50	71.88	2.91	59.18
LossDiff-IRM	64%	4.82	76.36	5.02	68.85	5.47	74.38	3.19	64.83
Pythia-410M (SLiC)									
SFT		2.56	-	2.47	-	3.15	-	1.91	-
Full Data	100%	2.81	73.80	2.82	73.91	3.31	71.25	2.03	57.49
Random	56%	2.67	69.23	2.69	70.68	3.09	63.12	2.14	59.77
GPT4	56%	2.80	72.75	2.85	74.53	3.23	75.62	2.15	60.07
Reward Model	56%	2.87	77.41	2.94	78.39	3.50	75.00	2.15	60.11
LossDiff-IRM	56%	3.07	80.08	3.09	84.39	3.83	79.36	2.21	62.40

Observation:

- **LossDiff-IRM achieves better performance over full-data training.**
- **LossDiff-IRM outperforms data selections based on GPT4 score or external reward model score.**

Comparisons & Ablation Study

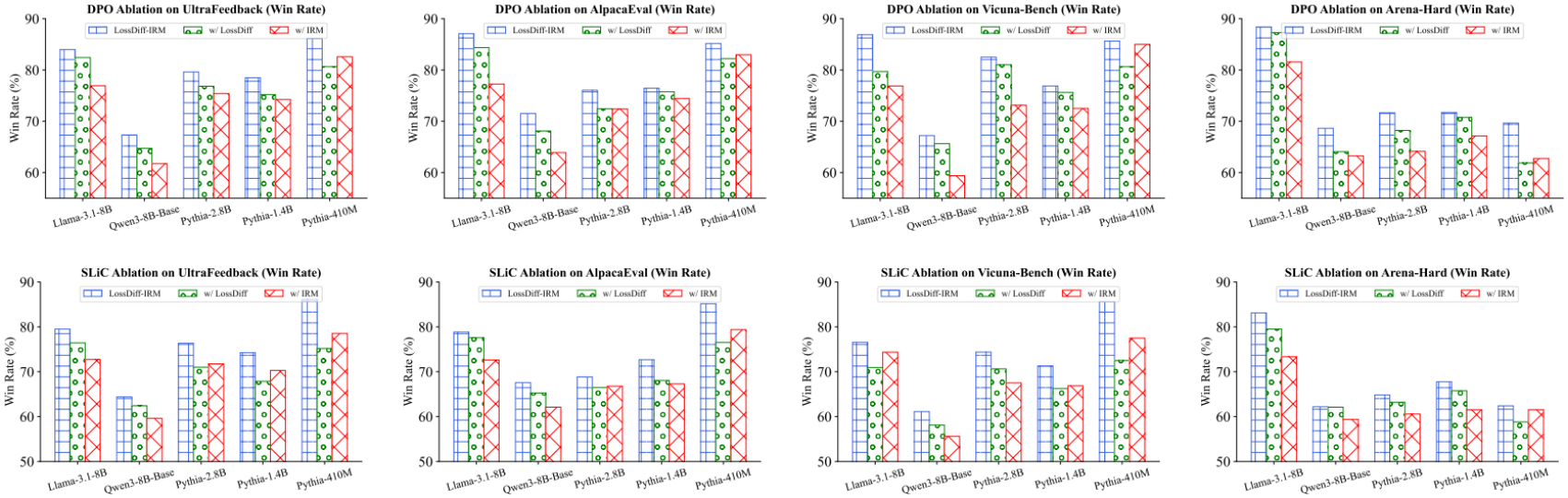
Training Data	UltraFeedback		AlpacaEval		Vicuna-Bench		Arena-Hard	
	Single ↑	WinRate ↑	Single ↑	WinRate ↑	Single ↑	WinRate ↑	Single ↑	WinRate ↑
Llama-3.1-8B (DPO)								
CurriDPO-GPT4	5.47	74.23	5.53	75.01	5.84	74.06	4.55	79.77
CurriDPO-Reward Model	5.51	74.62	5.54	74.29	5.59	74.06	4.62	79.49
M_{AP}	6.04	79.99	6.21	80.88	6.34	74.69	5.08	85.30
RS-DPO	5.70	75.98	6.39	84.84	7.04	82.81	4.69	82.05
LossDiff-IRM	6.54	83.97	6.84	87.08	7.06	86.88	5.59	88.40
Qwen3-8B-Base (DPO)								
CurriDPO-GPT4	7.61	61.04	7.74	62.35	8.16	54.69	7.52	62.51
CurriDPO-Reward Model	7.60	59.62	7.84	61.96	8.20	61.58	7.55	63.97
M_{AP}	7.95	67.84	8.31	71.11	8.62	66.87	7.72	66.55
RS-DPO	7.88	64.87	8.36	74.07	8.93	71.90	7.48	61.32
LossDiff-IRM	8.05	67.32	8.36	71.52	8.72	67.19	7.83	68.63

Observation:

- **LossDiff-IRM outperforms several data-centric baselines.**
- This indicates the effectiveness and superiority of our LossDiff-IRM selection criterion.

Observation:

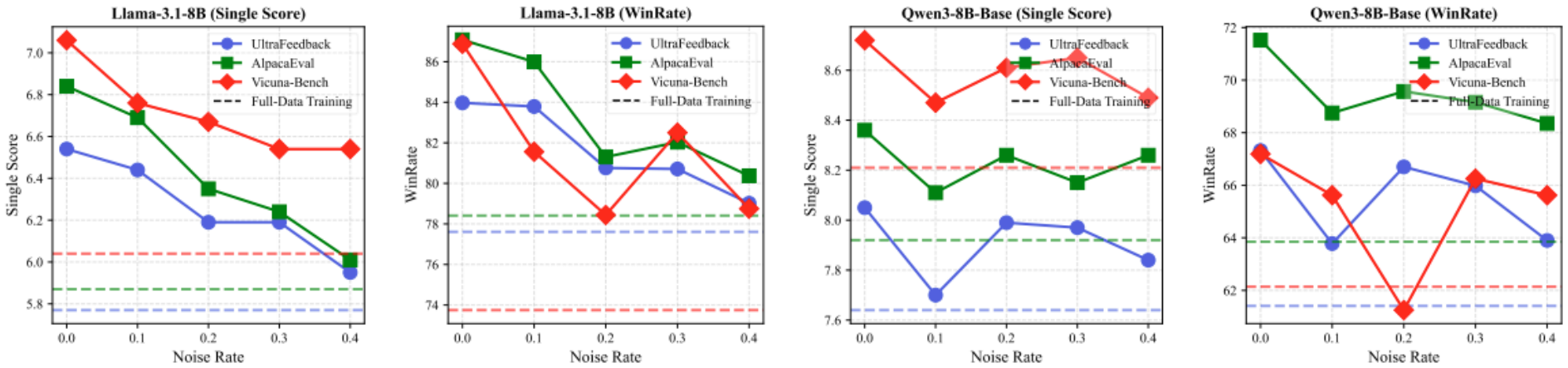
- **Combining LossDiff and IRM outperforms either alone.**
- Combining the LossDiff and IRM to offset specific errors each incurs when used alone to approximate TIF in data selection.



Further Analyses

Observation: Dropped data is low-value for the current model alignment.

Training Data	UltraFeedback		AlpacaEval		Vicuna-Bench		UltraFeedback		AlpacaEval		Vicuna-Bench	
	Single ↑	WinRate ↑	Single ↑	WinRate ↑	Single ↑	WinRate ↑	Single ↑	WinRate ↑	Single ↑	WinRate ↑	Single ↑	WinRate ↑
	Llama-3.1-8B (DPO)						*Qwen3-8B-Base (DPO)*					
- Full Data	5.77	77.61	5.87	78.41	6.04	73.75	7.64	61.41	7.92	63.85	8.21	62.14
- w/ Selected Data	**6.54**	**83.97**	**6.84**	**87.08**	**7.06**	**86.88**	**8.05**	**67.32**	**8.36**	**71.52**	**8.72**	**67.19**
- w/ Dropped Data	4.56	64.25	4.48	61.15	4.69	62.81	7.51	54.82	7.58	58.33	7.79	46.88
	Llama-3.1-8B (SLiC)						*Qwen3-8B-Base (SLiC)*					
- Full Data	5.09	70.72	5.13	72.13	5.40	71.88	7.55	59.54	7.61	59.71	8.05	54.69
- w/ Selected Data	**5.94**	**79.51**	**5.84**	**78.84**	**5.85**	**76.56**	**7.87**	**64.40**	**8.11**	**67.58**	**8.44**	**61.12**
- w/ Dropped Data	4.22	60.08	4.33	59.37	4.89	62.81	7.37	56.67	7.33	55.79	8.21	56.72



Observation: LossDiff-IRM exhibits a certain robustness under validation set noise.

Take-home Message

- The preference data quality is model-dependent.
- We investigate the valuable preference data from the perspective of generalization using influence function (IF).
- **The medium-IF preference data is more valuable** for alignment training.
- Computing exact influence functions is computationally expensive, so we propose **LossDiff-IRM as an efficient approximation that is highly correlated with exact IF**.
- We empirically demonstrate the effectiveness of LossDiff-IRM to achieve superior performance over full-data training, other data selection strategies, and several data-centric baseline.

Thank you!

If you have any question, feel free to contact me.

Zizhuo Zhang
cszzzhang@comp.hkbu.edu.hk