Findings of the Association for Computational Linguistics: NAACL 2024, pages 2723–2735

Contrastive Preference Learning for Neural Machine Translation

Jianfei He[1], **Shichao Sun**[2], **Sen Peng**[1], **Jie Xu**[1], **Xiaohua Jia**[1], **Wenjie Li**[2]
[1] City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong
[2] The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
jianfeihe-2c@my.cityu.edu.hk, bruce.sun@connect.polyu.hk
{senpeng.cs, jiexu49-c}@my.cityu.edu.hk
csjia@cityu.edu.hk, wenjie.li@polyu.edu.hk

Abstract

There exists a discrepancy between the token-level objective during training and the overall sequence-level quality that is expected from the model. This discrepancy leads to issues like exposure bias. To align the model with human expectations, sequence-level objectives are often used to fine-tune pre-trained models. In this paper, we introduce a contrastive preference model that enhances the traditional Plackett-Luce model by incorporating an indicator function. Building upon this novel preference model, we propose Contrastive Preference Learning (CPL), which uses offline samples with list-wise preferences to fine-tune a pre-trained model in Neural Machine Translation. Our experiments, conducted on three language pairs, demonstrate that CPL outperforms not only the vanilla Transformer model but also other token-level and sequence-level baselines. Furthermore, the ablation study highlights the essential role of the proposed indicator function in achieving this improvement.

1 Introduction

Neural Machine Translation (NMT) models (Bahdanau et al., 2014), like many other text generation tasks, are typically trained using *teacher forcing* and the token-level Maximum Likelihood Estimation (MLE) as the objective function. However, there exists a discrepancy between this training approach and the actual goal of a sequence generation system, which is to improve sequence-level quality as measured by evaluation metrics like BLEU, or human evaluation. One issue stemming from this disparity is *exposure bias* (Bengio et al., 2015; Ranzato et al., 2016; Wang and Sennrich, 2020; Korakakis and Vlachos, 2022). Wang and Sennrich (2020) also link this discrepancy to other issues observed in NMT: hallucination, domain shift, and beam search curse (Koehn and Knowles, 2017). This same discrepancy is the underlying reason behind the topic of *alignment* in Large Language Models (LLMs). Aligning LLMs with human expectations has been recognized as an important objective for future Artificial General Intelligence (AGI)[1], leading to an active research area (Wang et al., 2023). Approaches developed in both domains can be mutually beneficial.

To mitigate this discrepancy, sequence-level objectives are often used to fine-tune a pre-trained model (Edunov et al., 2018). There are two lines of related research: Reinforcement Learning (RL) with online samples and Supervised Learning with offline samples.

In the approach using RL with online samples, samples are refreshed by drawing from the model at every training step. This approach has been extensively discussed in NMT. MIXER (Ranzato et al., 2016) and Minimum Risk Training (MRT) (Shen et al., 2016) are two prominent implementations. However, there is still ongoing debate regarding the stability and effectiveness of these solutions (Choshen et al., 2020; Kiegeland and Kreutzer, 2021). In LLMs, the use of Reinforcement Learning with Human Feedback (RLHF) (Ouyang et al., 2022; Touvron et al., 2023) is common. The reward function for RL is trained using offline preference samples labeled by human experts, but online samples are still used during the RL phase. Methods utilizing online samples are significantly slower than offline ones.

The alternative approach is Supervised Learning with offline samples. These samples are drawn from the pre-trained model once and then ranked and used as training data to fine-tune the model with Supervised Learning. Unlike online samples, these offline samples are not refreshed during training. This method has been explored in

[1] https://openai.com/blog/introducing-superalignment

text summarization through *Contrastive Learning* with list-wise ranking (Sun and Li, 2021; Liu et al., 2022; Zhao et al., 2022). In LLMs, Direct Preference Optimization (DPO) [2] (Rafailov et al., 2023) uses a loss function which has been theoretically proven equivalent to RL with preference data. However, their experiments primarily focus on pair-wise preference rather than list-wise ranking.

Our proposal, *Contrastive Preference Learning* (CPL), follows the second approach since it is more efficient and stable compared to the RL with online samples. We begin by augmenting the classic list-wise Plackett-Luce (PL) preference model (Plackett, 1975; Luce, 1959) with an indicator function. This indicator function incorporates a constraint commonly used in contrastive learning to prevent overfitting. Then, the training objective is derived by applying the reward function in DPO to this augmented PL model. Our experiments consider three language pairs and compare CPL against various baselines. These baselines include the vanilla Transformer, Contrastive Learning and DPO with offline list-wise ranking, two online sequence-level methods (MIXER and MRT), and three token-level methods aimed at mitigating the exposure bias. The results show that CPL significantly outperforms the vanilla Transformer and achieves the best performance among all methods. The ablation study shows the crucial role played by the proposed indicator function.

2 Related Work

2.1 Exposure Bias and Alignment

The discrepancy between the training with teacher forcing and normal inference is well recognized in NMT. Exposure bias is often regarded (Bengio et al., 2015; Ranzato et al., 2016) as a consequence of this discrepancy. The existence of exposure bias has been proven by Wu et al. (2018) and Korakakis and Vlachos (2022) through the measurement of error accumulation. Wang and Sennrich (2020) provide indirect evidence for exposure bias with the experiments showing that MRT as a sequence-level objective can improve performance. Besides NMT, Chiang and Chen (2021) and Arora et al. (2022) quantify exposure bias in text completion.

In LLMs, this discrepancy often leads to a research topic known as *alignment*, which has been recognized as an important objective for the future Artificial General Intelligence (AGI). Wang et al. (2023) provide a comprehensive overview of alignment technologies.

2.2 Token-Level Approach

To mitigate the exposure bias, the token-level approach exposes the model to its predictions besides the ground truth. Scheduled Sampling (SS), introduced by Bengio et al. (2015), dynamically draws samples from the model's predictions and replaces the ground truth tokens. Mihaylova and Martins (2019) implement SS to the Transformer architecture (Vaswani et al., 2017). Additionally, Liu et al. (2021) propose *Confidence-Aware Scheduled Sampling (CASS)*, which improves the performance by selecting samples based on the log probability of the ground truth token. Furthermore, Goodman et al. (2020) introduce *TeaForN*, which utilizes a stack of decoders to update the model based on multiple prediction steps. There are some doubts about SS. Huszár (2015) proves that SS has an improper training objective. Some experiments (Mihaylova and Martins, 2019; Korakakis and Vlachos, 2022) show that SS performs worse than teacher forcing. These methods are implemented as baselines in our experiments.

2.3 Sequence-Level Approach

The sequence-level approach uses a sequence-level loss function and directly maximizes the total quality of the generated sequences. There are two categories of approaches.

One is Reinforcement Learning (RL) with online samples that are dynamically generated from the model during training. Ranzato et al. (2016) propose MIXER, which is based on a basic RL algorithm called REINFORCE. MRT (Shen et al., 2016; Wang and Sennrich, 2020) aims to minimize the expected discrepancy between the gold references and the model predictions. These online sampling methods need to generate samples token-by-token during training. According to Edunov et al. (2018), the online setting is 26 times slower than the corresponding offline setting. Furthermore, there has been some debate regarding these methods. Choshen et al. (2020) identify multiple weaknesses of MIXER and MRT, suspecting that they do not optimize the expected reward. However, Kiegeland and Kreutzer (2021) have provided empirical evidence contradicting these claims. In LLMs, RLHF (Ouyang et al., 2022)

[2] `https://huggingface.co/docs/trl/main/en/dpo_trainer`

uses samples ranked by human experts to align the output of LLMs with human intent. These preference samples are used to train a reward model, which is used by RL to fine-tune the LLM. This method has been widely used in LLMs such as LLama2 (Touvron et al., 2023).

The other approach is Supervised Learning with offline samples drawn from the pre-trained model before fine-tuning. There is a line of research that uses Contrastive Learning (CL) with *list-wise ranking* for the task of text summarization (Sun and Li, 2021; Liu et al., 2022; Zhao et al., 2022). In NMT, Edunov et al. (2018) introduce a margin loss in their comprehensive overview of classic sequence-level loss functions. This margin loss is similar to CL. However, they conduct experiments using a Recurrent Neural Network (RNN) instead of a Transformer. Another example using offline ranked samples in NMT is Lee et al. (2021). However, they use the ranking samples to train a separate reranking model in addition to the translation model, which incurs additional complexity and computation. Yang et al. (2019) and Pan et al. (2021) apply CL to NMT, but they address specific issues, namely word omission errors and interim presentation for many-to-many multilingual NMT, respectively. In LLMs, Rafailov et al. (2023) proposed Direct Preference Optimization (DPO) with a loss function that is theoretically equivalent to RL with offline preferences. Their solution and experiments mainly focus on the case of two preferences, i.e., good or bad. The loss function and its gradient are illustrated in Appendix A.

3 Preliminaries

3.1 The Plackett-Luce Model

The Plackett-Luce model is a preference model used to model list-wise ranking data and is widely used in list-wise Learning-To-Rank (LTR) methods (Cao et al., 2007; Xia et al., 2008; Ma et al., 2021) for building the ranking system. Let x be the input context, which is the source sentence in the case of NMT. Let $y_1, ..., y_K$ denote a set of K samples, and let τ be a permutation that represents the list-wise ranking of these samples. $\tau(k)$ refers to the *k-th* sample in the ranking, where a smaller k indicates a better sample. According to the Plackett-Luce model, the probability of observing a specific ordered list can be defined as follows:

$$p(\tau \mid y_1, ..., y_K, x) = \prod_{k=1}^{K-1} \frac{e^{S(x,y_k)}}{\Sigma_{j=k}^{K} e^{S(x,y_j)}}, \qquad (1)$$

where $S(x, y_k)$ is a utility score function. This function might be implicit in the case of human evaluation.

3.2 Contrastive Learning Using List-Wise Ranking

The key component in Contrastive Learning (CL) is the *max* function, defined as:

$$\max\{0, \rho + S_{negative} - S_{positive}\}, \qquad (2)$$

where $S_{negative}$ and $S_{positive}$ are scores for negative and positive samples, ρ is a hyperparameter for the margin.

The loss function for CL using list-wise ranking (Liu et al., 2022) is:

$$\mathcal{L}_{list}^{CL} = \sum_{k=1}^{K} \sum_{j=k+1}^{K} \max(0, \rho + log\, p_\theta(y_j \,|\, x) - log\, p_\theta(y_k \,|\, x)), \qquad (3)$$

where p_θ is the conditional probability of a sequence, $\rho = \lambda \,|\, k - j \,|$, λ is a hyperparameter.

This max function implies that when the score of the negative sample plus a margin is smaller than the score of the positive sample, the loss is zero.

4 Our Approach

4.1 Contrastive Preference Model

When directly using the PL model and maximizing the probability, the log probability of positive samples is *maximized*, while the log probability of negative samples is *minimized*. It occurs even when the positive samples already have higher probabilities than the negative samples, potentially leading to overfitting and conflicting with the requirements of other samples. To address this, we propose an augmented PL model that incorporates an indicator function, referred to as the *contrastive preference model*. The probability of observing a specific ordered list is defined as follows:

$$p_\theta(\tau \mid y_1, ..., y_K, x) = \prod_{k=1}^{K} \frac{e^{S(x,y_k)}}{\Sigma_{j=k}^{K} I(y_k, y_j, x) e^{S(x,y_j)}}, \qquad (4)$$

where the indicator function, which is inspired by contrastive learning, is defined as:

$$I(y_k, y_j, x) = \begin{cases} 0 & if \max(0, \rho + log\, p_\theta(y_j \mid x) - log\, p_\theta(y_k \mid x)) = 0, \\ 1 & otherwise \end{cases} \tag{5}$$

and $\rho = \lambda \mid k - j \mid$, λ is a hyperparameter.

Under this condition, the training objective shifts from maximizing the separation between the log probabilities of positive and negative samples to satisfying the margin requirement. Once the margin is met, this objective does not push the samples further apart. Theoretically, this approach prevents overfitting to samples that already satisfy the given conditions, allowing for parameter adjustments within the model to satisfy the requirements of other samples.

4.2 Contrastive Preference Learning

This section introduces *Contrastive Preference Learning* (CPL) as a novel approach based on the contrastive preference model. CPL aims to maximize the expected probabilities within the contrastive preference model. This optimization goal can be expressed equivalently as minimizing the following loss function:

$$\mathcal{L}_{list}^{CPL} = \\ - E_{(x,y_1,\ldots,y_k)\sim\mathcal{D}}[log \prod_{k=1}^{K} \frac{e^{S_\theta}(x, y_k)}{\sum_{j=k}^{K} I(y_k, y_j, x) e^{S_\theta(x, y_j)}}], \tag{6}$$

where $\mathcal{D}$ represents the training data set, and θ denotes the parameters of the model being trained.

One interesting candidate for the utility score function in Eq. 8 is the reward function derived by DPO. Their derivations reveal a surprising conclusion: optimizing a preference model with this reward function is theoretically equivalent to RLHF. This approach allows for bypassing the explicit reward modeling step and eliminates the need for performing reinforcement learning. The detailed derivations can be found in their paper (Rafailov et al., 2023).

The derived reward function is given by:

$$r_\theta(x, y) = \beta log \frac{p_\theta(y \mid x)}{p_{ref}(y \mid x)}, \tag{7}$$

where p_θ is the probability in the current model being trained and p_{ref} is the probability in the pre-trained model used to draw offline samples, β is a hyperparameter used as the weight of the implicit constraint term of the KL divergence.

By replacing the utility score function with this reward function, we obtain the loss function for CPL:

$$\mathcal{L}_{list}^{CPL}(p_\theta; p_{ref}) = \\ - E_{(x,y_1,\ldots,y_k)\sim\mathcal{D}}[log \prod_{k=1}^{K} \frac{e^{r_\theta(x, y_k)}}{\sum_{j=k}^{K} I(y_k, y_j, x) e^{r_\theta(x, y_j)}}]. \tag{8}$$

Since p_{ref} is independent of θ, we can compute the gradient for Eq. 7 as follows:

$$\nabla_\theta r_\theta(x, y_k) = \beta \nabla_\theta \log p_\theta(y_k \mid x). \tag{9}$$

Meanwhile,

$$e^{r_\theta(x, y_j)} = (\frac{p_\theta(y_j \mid x)}{p_{ref}(y_j \mid x)})^\beta. \tag{10}$$

So, the gradient for the CPL loss function is:

$$\nabla \mathcal{L}_{list}^{CPL}(p_\theta; p_{ref}) = \\ - E_{(x,y_1,\ldots,y_k)\sim\mathcal{D}} \sum_{k=1}^{K} [\beta \nabla_\theta \log p(y_k \mid x) - \\ \frac{\beta \sum_{j=k}^{K} I(y_k, y_j, x)(\frac{p_\theta(y_j|x)}{p_{ref}(y_j|x)})^\beta \nabla_\theta \log p(y_j \mid x)}{\sum_{j=k}^{K} I(y_k, y_j, x)(\frac{p_\theta(y_j|x)}{p_{ref}(y_j|x)})^\beta}]. \tag{11}$$

This equation offers insight into how the indicator function influences the training process. For each sample k in the ranking, its gradient component is determined by subtracting a *weighted average* of itself and the samples following it in the ranking from its own gradient. The weight assigned to each sample is defined by the exponential function of its implicit reward: $e^{r_\theta(x,y)} = (\frac{p_\theta(y|x)}{p_{ref}(y|x)})^\beta$. Without this indicator function, all negative samples contribute to the loss function, even if their probabilities are already smaller than their corresponding positive samples. It can result in the model overfitting to this specific list of samples. However, when the indicator function is included, these negative samples are excluded from the loss function, preventing overfitting and leaving space for optimization of other samples.

With some algebra, we can prove that the gradient of list-wise ranking with $K = 2$ and without the indicator function is equivalent to the pair-wise preference in DPO. This finding confirms the consistency of our derivation with DPO. The derivation is described in Appendix B.

4.3 Regularization Term

To prevent the finetuning model from deviating too much from the pre-trained model, we use a regularization term based on Cross Entropy (CE). We use the Negative Log-Likelihood (NLL) with Label Smoothing for this term (Edunov et al., 2018):

$$\mathcal{L}_{CE} = -\sum_{i=1}^{n} log\ p(y_i|x, y_{<i}) - D_{KL}(f \parallel p(y_i|x, y_{<i})), \tag{12}$$

where $f = \frac{1}{V}$ is uniform prior distribution over all tokens in the vocabulary with the size of V.

The loss function of CPL with this regularization term is

$$\mathcal{L}_{CPL} = \alpha \mathcal{L}_{list}^{CPL} + \mathcal{L}_{CE}. \tag{13}$$

4.4 Relation with DPO

Rafailov et al. (2023) discussed DPO with list-wise preference. The loss function is:

$$\mathcal{L}_{list}^{DPO}(p_\theta; p_{ref}) = \\ -E_{(x,y_1,\ldots,y_k)\sim\mathcal{D}}[log \prod_{k=1}^{K} \frac{e^{r_\theta(x,y_k)}}{\sum_{j=k}^{K} e^{r_\theta}(x, y_j)}]. \tag{14}$$

Comparing Eq. 14, Eq. 8, and Eq. 13, we can find that if we remove the indicator function and the regularization term, CPL is reduced to DPO with list-wise preference. Our experimental results show the significance of these two factors in achieving optimal system performance.

5 Experiments

5.1 Datasets

In our experiments, we use the corpora from WMT[3]. Wang and Sennrich (2020) claim that the methods reducing exposure bias with sequence-level objectives, such as MRT, can particularly enhance the model's resilience to domain shift. To evaluate this claim, we conduct Out-Of-Domain (OOD) tests on De–En and Ru–En language pairs.

For De–En, we use Europarl v7, News-commentary-v12, and Common Crawl for training (4.6 million sentences), Newstest2014 for validation, and Newstest2021 and EMEA[4] for in-domain and OOD testing respectively.

For Fr–En, we use Europarl v7, News-commentary-v10, and Common Crawl for training (5.4 million sentences), Newstest2013 for validation, and Newstest2014 for testing.

For Ru–En, we use ParaCrawl v9, News-commentary-v10, and Common Crawl for training(13.1 million sentences), Newstest2014 for validation, Newstest2021 for testing. The OOD tests for Ru–En use the test sets for the Biomedical Translation Task in WMT22[5].

These original datasets are first filtered. 350 million sentences are randomly selected with the conditions below:

- The length of source and target sentences are within the range of 5 to 300.
- The disparity between the length of the source and target sentences does not exceed five times.

To get the offline preference samples for finetuning, we use the pre-trained model to translate all training sentences. For each sentence, we generate eight n-best hypotheses based on their sequence probabilities. If all eight hypotheses receive BLEU scores lower than 15, we remove the corresponding sentence. The number of sentence pairs for each language pair is as follows: De–En 2.6 million, Ru–En 2.9 million, Fr–En 2.7 million.

We construct five preferences for each sentence in the filtered training data. The eight hypotheses from the pre-trained model are ranked according to their BLEU scores against the gold reference. We then choose the hypotheses with even orders (0, 2, 4, 6) as our list-wise ranked preferences. Besides, the reference sentence is always placed at the beginning of the list. In total, we generate five preferences for each sentence.

5.2 Systems

We implement Contrastive Preference Learning (CPL) described in Section 4.2. We use $\lambda = 0.1$ as Liu et al. (2022) and $\beta = 1$ as Rafailov et al. (2023).

We selected the weight α of CPL by monitoring the values of the loss components CE and CPL during training. We started from $\alpha = 1$ and found the value of CE loss increases in training, showing the deviation. Therefore, we selected an $\alpha < 1$ for CPL so that the CE term has a larger weight than CPL. We settled on $\alpha = 0.1$ since it worked well to justify the method.

[3] http://www.statmt.org

[4] https://opus.nlpl.eu/EMEA.php

[5] https://www.statmt.org/wmt22/biomedical-translation-task.html

	De–En (In-Domain)			De–En (OOD)			Ru–En (In-Domain)			Ru–En (OOD)		
Metrics	BLEU	Meteor	Comet	BLEU	Meteor	Comet	BLEU	Meteor	Comet	BLEU	Meteor	Comet
Baselines												
TX	31.29	49.68	76.60	25.86	41.64	67.95	30.35	49.68	75.07	35.01	51.93	75.06
SS	31.70	50.19	76.83	26.25	42.17	68.33	30.32	*49.54*	75.07	35.61	52.38	75.17
CASS	31.53	50.15	76.82	**26.76**	42.07	68.28	30.38	49.81	75.37	35.86	52.17	75.19
TFN	31.54	50.16	**77.11**	26.40	42.11	68.49	30.67	49.71	75.20	35.92	52.20	75.12
MIXER	31.71	50.15	76.83	26.62	42.23	68.47	*30.12*	*49.65*	75.35	35.60	52.19	75.34
MRT	31.37	50.11	**77.01**	26.40	42.11	68.20	30.32	*49.53*	75.16	**36.37**	52.81	75.29
CL	31.50	49.99	**76.88**	26.16	41.65	68.17	30.50	49.86	**75.42**	35.89	52.38	75.35
LDPO	*0.19*	*9.25*	*43.35*	*0.10*	*7.05*	*31.55*	*0.07*	*2.14*	*28.58*	*0.04*	*2.38*	*28.29*
Our Proposal												
CPL	**31.73**	**50.26**	**76.84**	**26.72**	**42.28**	**68.52**	**31.09**	**50.02**	**75.40**	**36.26**	**52.82**	**75.35**
Δ (-TX)	**0.44**	**0.58**	**0.24**	**0.86**	**0.64**	**0.57**	**0.74**	**0.34**	**0.33**	**1.25**	**0.89**	**0.29**
CPL w/o IF	31.37	50.01	76.73	26.29	41.86	68.32	30.91	50.00	75.19	35.75	52.57	75.43
Δ (-TX)	0.08	0.33	0.13	0.43	0.22	0.37	0.56	0.32	0.12	0.74	0.64	0.37

Table 1: Performance of different methods. The scores of CPL and those better than CPL are highlighted in **Bold**, while the scores that are worse than the vanilla Transformer (denoted as **TX**) are shown in *Italic*. Δ denotes the gain compared to TX.

	Fr–En		
Metrics	BLEU	Meteor	Comet
TX	35.00	53.01	78.76
SS	35.17	53.17	78.89
CASS	35.25	53.18	78.75
TFN	*34.97*	*52.99*	78.85
MIXER	*34.70*	*52.90*	*78.75*
MRT	*34.97*	53.18	78.84
CL	*34.99*	*52.89*	*78.71*
LDPO	*0.07*	*6.2*	*39.5*
CPL	**35.29**	**53.25**	**79.02**
Δ (-TX)	**0.29**	**0.24**	**0.26**
CPL w/o IF	*34.99*	*52.95*	78.81
Δ (-TX)	-0.01	-0.06	0.05

Table 2: Performance of different methods for Fr–En. The denotations are the same as in Figure 1.

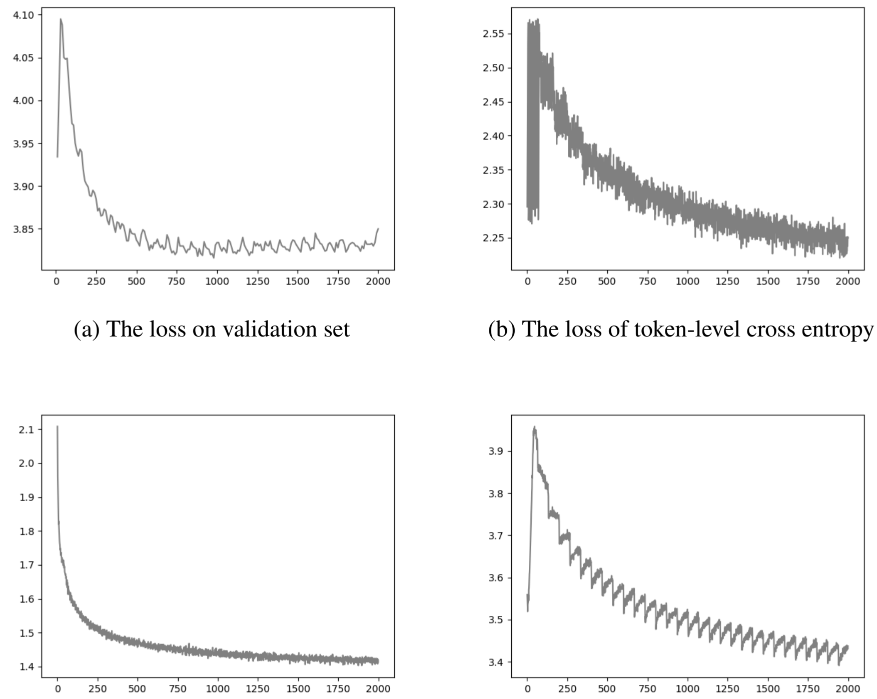
To conduct the ablation study, we implemented a variant of CPL without the indicator function. This system is denoted as *CPL w/o IF*.

We implement two methods using list-wise ranking that have not been explored in NMT to the best of our knowledge.

- *CL* is List-wise Contrastive Learning as described in Section 3.2. Its loss function includes the same regularization term as CPL: $\mathcal{L}_{CL} = \alpha\mathcal{L}_{list}^{CL} + \mathcal{L}_{CE}$.
- *LDPO* is list-wise DPO (Rafailov et al., 2023), which is defined by Eq. 14 in Section 4.4.

We compare our methods to the vanilla Transformer model and reimplement five methods introduced in Section 2 for comparison.

- *TX* is the vanilla Transformer.
- *SS* (Scheduled Sampling) (Mihaylova and Martins, 2019): We use Inverse Sigmod Decay for scheduling same as Liu et al. (2021).
- *CASS* (Confidence-Aware Scheduled Sampling) (Liu et al., 2021): We use its best configuration in their paper.
- *TFN* (Goodman et al., 2020): We use 0.4 as the second decoder's weight according to their recommendation.
- *MIXER* (Ranzato et al., 2016): Our implementation follows Kiegeland and Kreutzer (2021).
- *MRT* (Shen et al., 2016): We use four candidates and do not include the gold reference, same as Wang and Sennrich (2020).



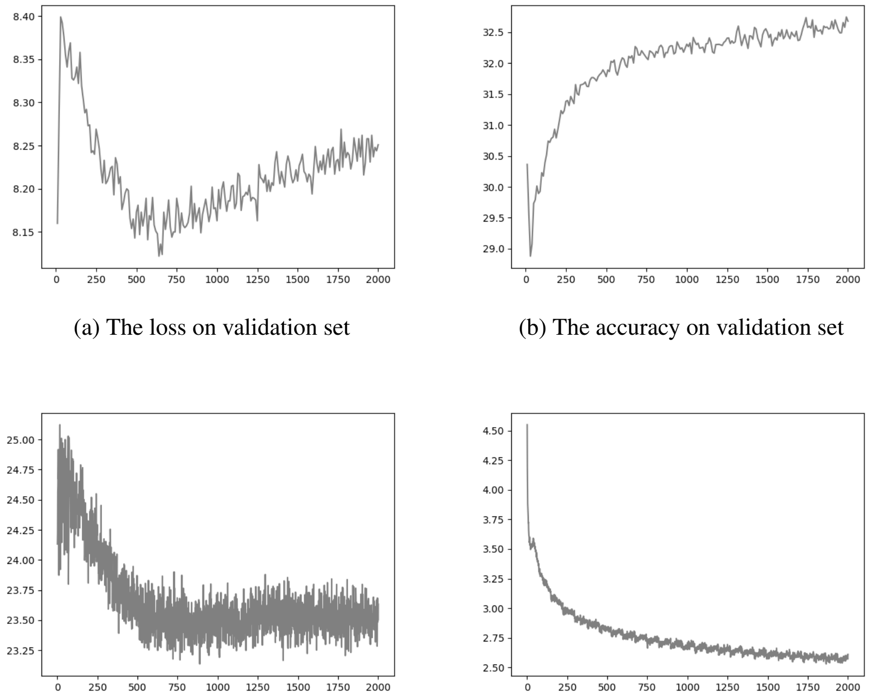
(a) The loss on validation set

(b) The loss of token-level cross entropy

(c) The loss of DPO

(d) The total loss

Figure 1: Investigate the components in the loss function for CPL for De–En for 30 epochs



(a) The loss on validation set

(b) The accuracy on validation set

(c) The loss of token-level cross entropy

(d) The loss of DPO

Figure 2: Investigate the components in the loss function of DPO only for De–En for 30 epochs

5.3 Implementation Details

Our implementation is based on the Fairseq toolkit (Ott et al., 2019) using a typical configuration [6] similar to the original Transformer (Vaswani et al., 2017). The Transformer Base model with about 60 million parameters is used. Both the dropout rate and the label smoothing are set to 0.1. We use the BPE (Sennrich et al., 2015) mode in SentencePiece[7] for subwords with 32,000 updates and use a shared vocabulary for source and target. Decoding is performed using beam search, with a beam size of five.

The pre-trained model is trained for a minimum of 20 epochs on the filtered data set described in Section 5.1, stopping if the validation loss does not decrease for 20 consecutive epochs. For fine-tuning, we adopt the same early-stop policy as Choshen et al. (2020), where the process is terminated if the validation loss does not decrease for ten consecutive epochs.

The CPL approach uses offline samples. As described in Section 2.3, using offline samples is 26 times slower than using online samples, according to Edunov et al. (2018). In our experiments, MIXER using online samples takes 25 minutes to finish 100 iterations, while the CPL with offline samples only takes 1.5 minutes, which is 17 times faster. The training speeds are close for all methods using offline samples, including list-wise contrastive learning and token-level methods such as SS, CASS, and TFN. For example, CASS takes one minute and ten seconds to finish 100 iterations, while the CPL takes 1.5 minutes.

5.4 Evaluation and Results

We evaluate the performance of the methods using three metrics: BLEU, Meteor, and Comet. For BLEU, We use SacreBLEU [8] (Post, 2018) [9]. Version 1.5 of Meteor [10] is used, and for Comet, we use the *wmt22-comet-da* model[11].

Table 1 illustrates the performance of methods for De–En and Ru–En.

The vanilla Transformer model is a strong baseline. For example, the experiments of Mihaylova and Martins (2019); Goodman et al. (2020) show very little gains in their experiments. Wang and Sennrich (2020) show gains in the out-of-domain tests but not on the in-domain tests.

Comparatively, CPL outperforms the vanilla Transformer model in all three metrics for all language pairs. It generally achieves the best performance when compared to other baselines. Additionally, the experiments on Out-of-domain tests show greater improvements than the in-domain tests. This result aligns with the conclusions of Wang and Sennrich (2020), suggesting that the exposure bias issue is more pronounced in out-of-domain scenarios. CPL using a sequence-level objective can alleviate this issue.

While CL with list-wise ranking also outperforms the vanilla Transformer model and demonstrates its efficacy in improving NMT, its gains are generally lower than CPL.

DPO with list-wise preference performs poorly in all tests, scoring below 0.2 in BLEU scores. The analysis in Section 2.3 illustrates its significant deviation from the pre-trained model, even when applying the highest weight value (5) for the KL divergence term, as mentioned in their study. Table 2 shows the performance of different methods for Fr–En, which gets consistent conclusions with the previous findings.

6 Analysis

6.1 Loss Components in CPL

Figure 1 shows the components in the loss function of CPL for De–En during training. Both the CPL loss component (Figure 1c) and the token-level cross entropy (Figure 1b) steadily decrease. These figures demonstrate the effectiveness of the CPL loss function presented in Section 4.2.

6.2 DPO Alone Deviates from the Pre-Trained Model

Figure 2 illustrates some information on training the DPO-only model for De–En. Figure 2a and Figure 2c demonstrate that the token-level loss on the validation set and on the train data CPL w/o IF significantly higher than expected during an effective training process, as illustrated in Figure 1a and Figure 1b for the CPL model. Furthermore, Figure 2b shows a much lower accuracy of around 30 compared to the typical accuracy of 60 or above achieved during training. These findings indicate that the DPO model deviates from the pre-trained model. Additionally, despite using a large weight

[6] https://github.com/facebookresearch/fairseq/tree/main/examples/scaling_nmt

[7] https://github.com/google/sentencepiece

[8] https://github.com/mjpost/sacreBLEU

[9] case.mixed+numrefs.1+smooth.exp+tok.13a+version.2.3.1

[10] http://www.cs.cmu.edu/~alavie/METEOR/

[11] https://github.com/Unbabel/COMET

β in the experiment, the implicit KL divergence term in DPO has no substantial effect.

6.3 Ablation Study

The ablation model CPL w/o IF is a variant of CPL, differing only in the absence of the indicator function. The results in Table 1 and Table 2 show that the improvements achieved by CPL w/o IF are considerably smaller than those of CPL. This finding highlights the significance of the indicator function in our proposed *contrastive preference model* and *contrastive preference learning*.

7 Conclusion

Using the sequence-level objective to fine-tune a pre-trained model is a promising way to align the model, trained with a token-level objective, with human expectations for high sequence-level quality. We augment the classic Plackett-Luce model with an indicator function. Based on this novel *contrastive preference model*, we propose *Contrastive Preference Learning* (CPL), which uses offline samples with list-wise preference to fine-tune a pre-trained model. Our experiments on three language pairs demonstrate that CPL outperforms the vanilla Transformer model and other token-level and sequence-level baselines. The proposed indicator function applies a constraint used in contrastive learning to prevent overfitting. Its crucial role is demonstrated in our ablation study.

Limitations

One limitation of this study is the influence of batch size on performance. Increasing the batch size has the potential to improve contrastive learning (Chen et al.). However, due to the limited memory capacity of our GPUs, we used a maximum batch size of 6000 tokens. Therefore, the impact of larger batch sizes was not extensively investigated in this study.

References

Kushal Arora, Layla El Asri, Hareesh Bahuleyan, and Jackie Chi Kit Cheung. 2022. Why exposure bias matters: An imitation learning perspective of error accumulation in language generation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 700–710.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136.

Changyou Chen, Jianyi Zhang, Yi Xu, Liqun Chen, Jiali Duan, Yiran Chen, Son Dinh Tran, Belinda Zeng, and Trishul Chilimbi. Why do we need large batchsizes in contrastive learning? a gradient-bias perspective. In *Advances in Neural Information Processing Systems*.

Ting-Rui Chiang and Yun-Nung Chen. 2021. Relating neural text degeneration to exposure bias. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 228–239.

Leshem Choshen, Lior Fox, Zohar Aizenbud, and Omri Abend. 2020. On the weaknesses of reinforcement learning for neural machine translation. In *International Conference on Learning Representations*.

Sergey Edunov, Myle Ott, Michael Auli, David Grangier, and MarcAurelio Ranzato. 2018. Classical structured prediction losses for sequence to sequence learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 355–364.

Sebastian Goodman, Nan Ding, and Radu Soricut. 2020. Teaforn: Teacher-forcing with n-grams. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8704–8717.

Ferenc Huszár. 2015. How (not) to train your generative model: Scheduled sampling, likelihood, adversary? *arXiv preprint arXiv:1511.05101*.

Samuel Kiegeland and Julia Kreutzer. 2021. Revisiting the weaknesses of reinforcement learning for neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1673–1681.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Michalis Korakakis and Andreas Vlachos. 2022. Improving scheduled sampling with elastic weight consolidation for neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7247–7258, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ann Lee, Michael Auli, and Marc'Aurelio Ranzato. 2021. Discriminative reranking for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7250–7264, Online. Association for Computational Linguistics.

Yijin Liu, Fandong Meng, Yufeng Chen, Jinan Xu, and Jie Zhou. 2021. Confidence-aware scheduled sampling for neural machine translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2327–2337.

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022. Brio: Bringing order to abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2890–2903.

R Duncan Luce. 1959. Individual choice behavior.

Jiaqi Ma, Xinyang Yi, Weijing Tang, Zhe Zhao, Lichan Hong, Ed Chi, and Qiaozhu Mei. 2021. Learning-to-rank with partitioned preference: Fast estimation for the plackett-luce model. In *International Conference on Artificial Intelligence and Statistics*, pages 928–936. PMLR.

Tsvetomila Mihaylova and André FT Martins. 2019. Scheduled sampling for transformers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 351–356.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. Contrastive learning for many-to-many multilingual neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 244–258.

Robin L Plackett. 1975. The analysis of permutations. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 24(2):193–202.

Matt Post. 2018. A call for clarity in reporting bleu scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

MarcAurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1692.

Shichao Sun and Wenjie Li. 2021. Alleviating exposure bias via contrastive learning for abstractive text summarization. *arXiv preprint arXiv:2108.11846*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc.

Chaojun Wang and Rico Sennrich. 2020. On exposure bias, hallucination and domain shift in neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3544–3552, Online. Association for Computational Linguistics.

Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. 2023. Aligning large language models with human: A survey. *arXiv preprint arXiv:2307.12966*.

Lijun Wu, Xu Tan, Di He, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2018. Beyond error propagation in neural machine translation: Characteristics of language also matter. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3602–3611.

Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. 2008. Listwise approach to learning to rank: theory and algorithm. In *Proceedings of the 25th international conference on Machine learning*, pages 1192–1199.

Zonghan Yang, Yong Cheng, Yang Liu, and Maosong Sun. 2019. Reducing word omission errors in neural machine translation: A contrastive learning approach. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6191–6196.

Yao Zhao, Mikhail Khalman, Rishabh Joshi, Shashi Narayan, Mohammad Saleh, and Peter J Liu. 2022. Calibrating sequence likelihood improves conditional language generation. In *The Eleventh International Conference on Learning Representations*.

A DPO for Pair-Wise Preference

The loss function of DPO for pair-wise preference is as follows:

$$\begin{aligned}&\mathcal{L}_{DPO}(p_\theta; p_{ref}) = -E_{(x,y_w,y_l)\sim\mathcal{D}}\\&[log\sigma(\beta\log\frac{p_\theta(y_w \mid x)}{p_{ref}(y_w \mid x)} - \beta\log\frac{p_\theta(y_l \mid x)}{p_{ref}(y_l \mid x)})],\end{aligned}$$

where y_w and y_l are positive samples (*win*) and negative samples (*lose*) in preferences.

Its gradient with respect to the parameters θ is

$$\begin{aligned}&\nabla_\theta\mathcal{L}_{DPO}(p_\theta; p_{ref}) = -\beta E_{(x,y_w,y_l)\sim\mathcal{D}}[\sigma(r_\theta(x, y_l)\\&- r_\theta(x, y_w))[\nabla_\theta logp(y_w \mid x) - \nabla_\theta logp(y_l \mid x)]],\end{aligned} \tag{15}$$

where,

$$\sigma(x) = \frac{1}{1+e^{-x}}, r_\theta(x, y) = \beta log\frac{p_\theta(y \mid x)}{p_{ref}(y \mid x)}. \tag{16}$$

The weight term $\sigma(r_\theta(x, y_l) - r_\theta(x, y_w))$ can be reformulated as

$$\begin{aligned}&\frac{1}{1+z^{-\beta}},\\&z = \frac{\frac{p_\theta(y_l|x)}{p_{ref}(y_l|x)}}{\frac{p_\theta y_w|x}{p_{ref}(y_w|x)}} = \frac{p_\theta(y_l \mid x)}{p_\theta(y_w \mid x)} \cdot \frac{p_{ref}(y_w \mid x)}{p_{ref}(y_l \mid x)}.\end{aligned} \tag{17}$$

This shows that the weight term in the gradient of PDO is determined by the relative change in sequence probability between the positive sample y_w and the negative sample y_l in both the current model and the pre-trained model. Behind the DPO's objective, they use the Bradley-Terry model, which calculates the probability of preference with a reward function:

$$\begin{aligned}p(y_1 \succ y_2 \mid x) &= \frac{e^{r(x,y_1)}}{e^{r(x,y_1)} + e^{r(x,y_2)}}\\&= \sigma(r(x, y_2) - r(x, y_1))\end{aligned} \tag{18}$$

B Gradience of CPL with List-Wise Ranking Reduced to DPO with Pair-Wise Preference

A list-wise ranking with only two samples is reduced to a pair-wise preference. The following derivation proves that the gradient of a special CPL (Eq. 11) with $K = 2$ and without the indicator function is equivalent to the pair-wise DPO in Eq. 15:

$$\begin{aligned}
\nabla\mathcal{L}_{DPO}(p_\theta; p_{ref}) &= -E_{(x,y_1,y_2)\sim\mathcal{D}}\sum_{k=1}^{2}[\beta\nabla_\theta \log p(y_k \mid x) - \frac{\beta\sum_{j=k}^{2}(\frac{p_\theta(y_j|x)}{p_{ref}(y_j|x)})^\beta \nabla_\theta \log p_\theta(y_j \mid x)}{\sum_{j=k}^{2}(\frac{p_\theta(y_j|x)}{p_{ref}(y_j|x)})^\beta}] \\
&= -E_{(x,y_1,y_2)\sim\mathcal{D}}[\beta\nabla_\theta \log p_\theta(y_1 \mid x) + \beta\nabla_\theta \log p_\theta(y_2 \mid x) - \frac{\beta\sum_{j=1}^{2}(\frac{p_\theta(y_j|x)}{p_{ref}(y_j|x)})^\beta \nabla_\theta \log p_\theta(y_j \mid x)}{\sum_{j=1}^{2}(\frac{p_\theta(y_j|x)}{p_{ref}(y_j|x)})^\beta}] - \beta\nabla_\theta \log p_\theta(y_2 \mid x)] \\
&= -E_{(x,y_1,y_2)\sim\mathcal{D}}[\beta\nabla_\theta \log p_\theta(y_1 \mid x) - \frac{\beta(\frac{p_\theta(y_1|x)}{p_{ref}(y_1|x)})^\beta \nabla_\theta \log p_\theta(y_1 \mid x) + \beta(\frac{p_\theta(y_2|x)}{p_{ref}(y_2|x)})^\beta \nabla_\theta \log p_\theta(y_2 \mid x)}{(\frac{p_\theta(y_1|x)}{p_{ref}(y_1|x)})^\beta + (\frac{p_\theta(y_2|x)}{p_{ref}(y_2|x)})^\beta}] \\
&= -E_{(x,y_1,y_2)\sim\mathcal{D}}\beta[\frac{(\frac{p_\theta(y_2|x)}{p_{ref}(y_2|x)})^\beta(\nabla_\theta \log p_\theta(y_1 \mid x) - \nabla_\theta \log p_\theta(y_2 \mid x))}{(\frac{p_\theta(y_1|x)}{p_{ref}(y_1|x)})^\beta + (\frac{p_\theta(y_2|x)}{p_{ref}(y_2|x)})^\beta}] \\
&= -\beta E_{(x,y_1,y_2)\sim\mathcal{D}}[\frac{\nabla_\theta \log p_\theta(y_1 \mid x) - \nabla_\theta \log p_\theta(y_2 \mid x)}{1 + \frac{(\frac{p_\theta(y_1|x)}{p_{ref}(y_1|x)})^\beta}{(\frac{p_\theta(y_2|x)}{p_{ref}(y_2|x)})^\beta}}] = -\beta E_{(x,y_1,y_2)\sim\mathcal{D}}[\frac{\nabla_\theta \log p_\theta(y_1 \mid x) - \nabla_\theta \log p_\theta(y_2 \mid x)}{1 + e^{\log \frac{(\frac{p_\theta(y_1|x)}{p_{ref}(y_1|x)})^\beta}{(\frac{p_\theta(y_2|x)}{p_{ref}(y_2|x)})^\beta}}}] \\
&= -\beta E_{(x,y_1,y_2)\sim\mathcal{D}}[\frac{\nabla_\theta \log p_\theta(y_1 \mid x) - \nabla_\theta \log p_\theta(y_2 \mid x)}{1 + e^{-(\beta \log \frac{p_\theta(y_2|x)}{p_{ref}(y_2|x)} - \beta \log \frac{p_\theta(y_1|x)}{p_{ref}(y_1|x)})}}] \\
&= -\beta E_{(x,y_1,y_2)\sim\mathcal{D}}[\sigma(r_\theta(x,y_2) - r_\theta(x,y_1))[\nabla_\theta logp_\theta(y_1 \mid x) - \nabla_\theta logp_\theta(y_2 \mid x)]].
\end{aligned} \tag{19}$$