CONGRAD: Conflicting Gradient Filtering for Multilingual Preference Alignment

Jiangnan Li[1], Thuy-Trang Vu[1], Christian Herold[2], Amirhossein Tebbifakhr[2], Shahram Khadivi[2], Gholamreza Haffari[1]
[1]Department of Data Science and AI, Monash University, Australia
[2]eBay Inc.
{first.last, trang.vu1}@monash.edu,
{cherold, atebbifakhr, skhadivi}@ebay.com

Abstract

Naive joint training of large language models (LLMs) for multilingual preference alignment can suffer from *negative interference*. This is a known issue in *multilingual training*, where conflicting objectives degrade overall performance. However, the impact of this phenomenon in the context of *multilingual preference alignment* remains largely underexplored. To address this issue, we propose CONGRAD, an effective and scalable filtering method that mitigates this interference by identifying and selecting preference samples that exhibit high cross-lingual affinity. Based on principles of multi-objective optimization, our approach computes an aggregated, cross-lingually beneficial gradient direction and uses this to filter for samples whose individual gradients align with this consensus direction. To ensure scalability for LLMs, we incorporate a *sublinear gradient compression* strategy that reduces memory overhead during gradient accumulation. We integrate CONGRAD into a self-rewarding framework and evaluate on `LLaMA3-8B` and `Gemma2-2B` across 10 languages. Results show that CONGRAD consistently outperforms strong baselines in both seen and unseen languages, with minimal alignment tax. [1]

1 Introduction

Preference alignment has emerged as a pivotal post-training technique for aligning large language models (LLMs) with human values and intentions (Ouyang et al., 2022; Touvron et al., 2023). It has driven significant performance improvements across multiple natural language processing (NLP) tasks such as summarisation (Ziegler et al., 2019), reasoning (Pang et al., 2024; Chen et al., 2024d), and instruction following (Bai et al., 2022; Ouyang et al., 2022). However, the progress in preference alignment research remains predominantly English-centric due to the scarcity of high-quality human preference data for non-English languages and the prohibitive costs of human annotation. Previous research observes that state-of-the-art aligned LLMs often overfit to English capabilities with noticeable degraded alignment and odd behaviours in less-represented languages (Schwartz et al., 2022; Kotek et al., 2023; Khondaker et al., 2023; Vashishtha et al., 2023; Deng et al., 2023).

To circumvent the need for costly human annotation in less-represented languages, recent efforts have turned to synthetic data generation (Hurst et al., 2024), often using powerful but proprietary models such as `OpenAI GPT-4o` (Hurst et al., 2024) as judges to score responses (Dubois et al., 2023; Lee et al., 2024; Cui et al., 2024) or employing self-rewarding methods where a model iteratively generates and evaluates its own outputs (Yuan et al., 2024). While promising, these strategies frequently rely on translating English-centric datasets (Lai et al., 2023; Chen et al., 2024c), a process that can introduce subtle artifacts and fails to generate diverse samples, which are important to robust model performances (Kirk et al., 2024). Once constructed, these multilingual preference datasets are typically used to jointly train aligned multilingual LLMs (Dang et al., 2024).

Nevertheless, it is well-known in multilingual machine translation and pre-training that naive optimization in multilingual tasks often exhibits *negative interference*, where conflicting per-language objectives often lead to sub-optimal performance (Wang et al., 2020). Similar to multilingual machine translation, multilingual preference alignment is also fundamentally a multi-objective learning problem and prone to the negative interference issue (Wang et al., 2020, 2021). While this issue has been widely studied in multilingual machine translation and pretraining (Arivazhagan et al., 2019; Wang et al., 2020; Conneau et al., 2020; Wu et al., 2021; Choi et al., 2023; Wu et al., 2024a),

[1] Code will be released on https://github.com/KagamiBaka/CONGRAD.

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics Volume 1: Long Papers, pages 6371–6387
March 24-29, 2026 ©2026 Association for Computational Linguistics

the impact of negative interference in multilingual preference alignment remains largely unexplored.

This paper directly addresses this gap. We propose ConGrad (CONflicting GRADient filtering), an effective and scalable method to mitigate negative interference in multilingual preference alignment. Our approach is motivated by the insight that gradient conflicts are strong indicators of task interference (Sener and Koltun, 2018; Yu et al., 2020). ConGrad filters the training data by identifying and retaining only those preference samples whose gradients align with a consensus direction beneficial to all languages to alleviate negative interference. This is achieved by first aggregating accumulated exponential moving average (EMA) gradients across languages and resolving their conflicts, and then selecting the top-k samples with the highest similarity to this de-conflicted, consensus gradient. Unlike traditional algorithms that directly modify gradients (Sener and Koltun, 2018; Yu et al., 2020; Wang et al., 2021), our method filters data samples to influence optimization indirectly. This approach is better suited for LLMs as it avoids the significant memory/computational overhead and potential training instability associated with direct gradient manipulation (Kurin et al., 2022). Besides, to ensure this method is more scalable for contemporary LLMs, we incorporate a sublinear gradient compression strategy based on subspace iteration (Bathe and Wilson, 1972) that makes storing and processing gradients memory-efficient.

We integrate our proposed ConGrad filtering method into the self-rewarding framework (Yuan et al., 2024) with Direct Preference Optimization (DPO) training (Rafailov et al., 2023). To evaluate its effectiveness, we conduct experiments on two LLMs of different scales: `Llama3-8B` (Grattafiori et al., 2024) and `Gemma2-2B` (Team et al., 2024) across 10 languages from different language families and writing scripts. The experimental results demonstrate the importance of high-quality preference data, where most filtering methods outperform the baselines without data filtering. Our proposed ConGrad method consistently outperforms strong filtering baselines on the multilingual instruction following benchmark for both languages seen during training, generalised well to unseen languages and low-resource languages, without significant alignment tax (Lin et al., 2024).

In summary, our contributions are as follows.

- We propose a conflict-aware self-rewarding algorithm for multilingual preference alignment, which constructs preference data without relying on external annotations and enables iterative self-improvement across languages.
- We introduce a gradient-level filtering strategy based on PCGrad that selects preference samples with high cross-lingual affinity, effectively mitigating negative interference during multilingual training.
- We conduct comprehensive experiments on `Llama3-8B` and `Gemma2-2B`, showing our method significantly improves underrepresented language while preserving dominant language capabilities and generalizes to unseen languages with minimal alignment tax.

2 Preliminaries

Multilingual Preference Alignment In this paper, we study the problem of multilingual preference alignment. For a given set of languages L, each language $l \in L$ is associated with a preference dataset $\mathcal{D}^l = \{(x_i^l, y_{i,c}^l, y_{i,r}^l)\}_{i=1}^{N^l}$ where N^l is the dataset size. Each data point is a tuple of prompt input x_i^l, chosen response $y_{i,c}^l$ and rejected response $y_{i,r}^l$. For brevity, we omit the index i when it is clear from the context.

The goal of mono-lingual alignment is to align the LLM $\mathcal{M}_\theta$, parameterised by θ, using preference optimisation methods such as reinforcement learning with human feedback (RLHF), *e.g.*, PPO (Schulman et al., 2017) or Direct Preference Optimisation (DPO) (Rafailov et al., 2023). We adopt DPO due to its simplicity and strong empirical performance. DPO directly minimises the negative log probability of preferring the chosen response y_c over the rejected response y_r given prompt x,

$$\mathcal{L}_{\text{DPO}}(\mathcal{M}_\theta, \mathcal{D}) = -\mathbb{E}_{(x,y_c,y_r)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\mathcal{M}_\theta(y_c|x)}{\mathcal{M}_{\text{ref}}(y_c|x)} - \beta\log\frac{\mathcal{M}_\theta(y_r|x)}{\mathcal{M}_{\text{ref}}(y_r|x)}\right)\right] \quad (1)$$

where $\mathcal{M}_{\text{ref}}$ is the reference model. When it extends to multilingual, we jointly optimise the DPO loss across multiple languages,

$$\mathcal{L}_{\text{mpa}}(\mathcal{M}_\theta) = \frac{1}{|L|}\sum_{l\in L}\mathcal{L}_{\text{DPO}}(\mathcal{M}_\theta, \mathcal{D}^l). \quad (2)$$

Compared to mono-lingual alignment, alignment in a multilingual context presents additional challenges, such as negative inter-language interference (Dang et al., 2024; Yu et al., 2020). The objective of this paper is to mitigate this problem by designing a data filtering algorithm to filter data that is prone to negative inter-language interference.

Self-rewarding Iterative DPO Collecting human preferences for training LLMs is a resource-intensive task. Self-rewarding method addresses this challenge by leveraging LLMs to generate responses and evaluate them to construct their own training preference data (Yuan et al., 2024).

Specifically, starting from an instruction-tuned model $\mathcal{M}_{\theta_0}$, we apply an iterative DPO training procedure. At iteration $t > 1$, the model $\mathcal{M}_{\theta_t}$ is initialised with the previous model $\mathcal{M}_{\theta_{t-1}}$. For each prompt x^l, it generates a set of k candidate responses $\{y_1^l, \ldots, y_k^l\}$. The same model $\mathcal{M}_{\theta_{t-1}}$ is then used to evaluate each response, resulting in reward scores $\{r_1^l, \ldots, r_k^l\}$. Since the model has the strongest instruction-following ability in English, we use the same English prompt when scoring responses in all languages $l \in L$. Preference pairs are then constructed by selecting the highest and lowest-scoring responses, discarding pairs with identical scores. These pairs are then used to train $\mathcal{M}_{\theta_t}$. Further details of the prompts are provided in Appendix D.

3 Conflicting Gradient Filtering for Multilingual Preference Alignment

To address the challenge of conflicting cross-lingual preferences in multilingual alignment, we introduce Conflicting Gradient Filtering (ConGrad), a novel framework designed to select high-quality preference data by minimizing gradient conflicts across languages. Our core idea is to first identify a consensus update direction that is beneficial for all languages and then filter self-generated preference pairs to align with this direction. This ensures a more coherent and effective multilingual optimization trajectory. As illustrated in Figure 1, ConGrad integrates seamlessly into the self-rewarding loop and comprises two key stages: deriving a consensus gradient and using it for efficient preference data filtering.

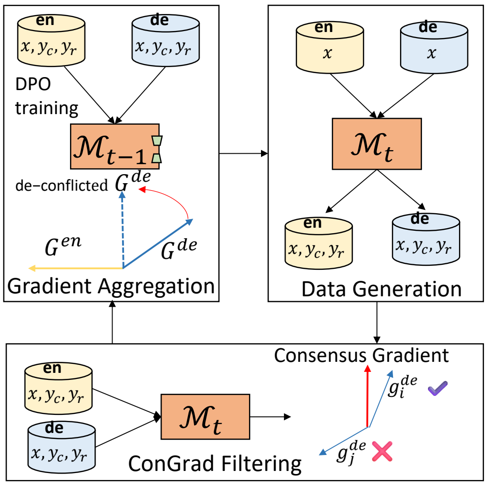

Figure 1: Overview of the multilingual preference alignment framework which consists of three steps: (i) computation of the cross-lingual consensus gradient via PCGrad with gradient compression for memory efficiency; (ii) generation of synthetic preference data via self-rewarding; and (iii) ConGrad filtering, which selects preference samples based on their gradient similarity to the consensus gradient direction, retaining only those with high similarity ($\mathbf{g}_i^{de}$ in this example).

3.1 Deriving a Consensus Gradient

Our primary objective is to find a single gradient update direction that harmonizes the learning objectives across multiple languages. Such a consensus direction should ideally capture shared preferences between languages while mitigating the negative interference caused by conflicting language-specific gradients. Wang et al. (2021) previously observed that higher gradient similarity correlates with improved multilingual performance. Building on this, our goal is to construct a gradient that captures this cross-lingual consensus.

Consensus Gradient Computation To achieve this, we first need a stable representation of each language's update direction within a given training iteration t. We compute the Exponential Moving Average (EMA) of gradients, G^l, for each language l over its minibatches $b_\tau^l \sim \mathcal{D}_t^l$ (see Algorithm 1). The EMA gradient is calculated as $G_\tau^l = \gamma G_{\tau-1}^l + (1-\gamma)\mathbf{g}_\tau^l$, where $\mathbf{g}_\tau^l$ is the minibatch gradient and γ is a decay factor.

With these stable language-specific gradients, we then need a mechanism to resolve their conflicts. We adapt the principles from Projecting Conflict Gradient (PCGrad) (Yu et al., 2020) for this purpose. For each language l, we initialize a candidate gradient $\mathbf{g}_{\text{pc}}^l$ with its EMA gradient G^l. We then iteratively project $\mathbf{g}_{\text{pc}}^l$ away from the conflicting components of other languages' gradients ($G^{l'}$). A conflict is identified when their

Algorithm 1 Conflicting Gradient Filtering

```
Require: Set of languages $L$, gradient EMAs $\{G^l\}_{l \in L}$, self-rewarding dataset $\{\mathcal{D}^l\}_{l \in L}$ , model $\mathcal{M}_\theta$
Initialise $\mathbf{g}^l_{\text{pc}} \leftarrow G^l \quad \forall l \in L$
for each language $l \in L$ do
    for each language $l' \in \text{shuffle}(L \setminus \{l\})$ do
        if $\mathbf{g}^l_{\text{pc}} . G^{l'} < 0$ then
            $\mathbf{g}^l_{\text{pc}} \leftarrow \mathbf{g}^l_{\text{pc}} - \frac{\mathbf{g}^l_{\text{pc}} \cdot G^{l'}}{\|G^{l'}\|^2} G^{l'}$
        end if
    end for
end for
$\mathbf{g}_{\text{pc}} = \sum_{l \in L} \mathbf{g}^l_{\text{pc}}$                // consensus gradient
// Filtering step
for each language $l \in L$ do
    for each sample $(x^l_i, y^l_{i,c}, y^l_{i,r}) \in \mathcal{D}^l$ do
        $\mathbf{g}^l_i \leftarrow \nabla_\theta \mathcal{L}_{\text{LP-DPO}}(\mathcal{M}_\theta, (x^l_i, y^l_{i,c}, y^l_{i,r}))$
    end for
    $\mathcal{S}^l_{\text{selected}} \leftarrow \text{ArgTopK}_{(x^l_i, y^l_{i,c}, y^l_{i,r}) \in \mathcal{D}^l} \cos(\mathbf{g}^l_i, \mathbf{g}_{\text{pc}})$
end for
return $\{\mathcal{S}^l_{\text{selected}}\}_{l \in L}$
```

Algorithm 2 Modified Iterative Self-Rewarding with ConGrad Filtering

```
Require: Set of languages $L$, seed model $\mathcal{M}_{\theta_0}$, prompt dataset $\{\mathcal{X}^l\}_{l \in L}$, total iterations $T$
initialise $\tilde{G}^l \leftarrow 0$, for all $l \in L$
for $t = 1$ to $T$ do
    for each language $l \in L$ do
        gen responses $\mathcal{Y}^l$ using $\mathcal{X}^l$ and $\mathcal{M}_{\theta_{t-1}}$
        gen rewards $\mathcal{R}^l$ for $\mathcal{Y}^l$ via self-eval
        construct $\mathcal{D}^l_t$ via pref pairs in $\mathcal{Y}^l$ and rewards $\mathcal{R}^l$
    end for
    if $t > 1$ then
        $\{S^l_{\text{selected}}\}_{l \in L} \leftarrow$
        ALGORITHM 1$(L, \{\tilde{G}^l * P\}_{l \in L}, \{\mathcal{D}^l_t\}_{l \in L}, \mathcal{M}_{\theta_{t-1}})$
    else
        $\{S^l_{\text{selected}}\}_{l \in L} \leftarrow \{\mathcal{D}^l_t\}_{l \in L}$
    end if
    $\mathcal{M}_{\theta_t} \leftarrow \mathcal{M}_{\theta_{t-1}}$
    for minibatches $b^l_\tau \in \{S^l_{\text{selected}}\}_{l \in L}$ do
        $\mathbf{g}^l_\tau \leftarrow \nabla_{\theta_\tau} \mathcal{L}_{\text{LP-DPO}}(\mathcal{M}_\theta, b^l_\tau)$
        $\mathcal{M}_{\theta_t} \leftarrow \text{UpdateModel}(\mathcal{M}_{\theta_t}, \mathbf{g}^l_\tau)$
        $G^l_{\text{old}} \leftarrow P^l_{\text{ema}}(Q^l_{\text{ema}})^\top$ {Decompress}
        $G'_{\text{updated}} \leftarrow \gamma G^l_{\text{old}} + (1 - \gamma)\mathbf{g}^l_\tau$ {Update}
        $P^l_{\text{ema}}, Q^l_{\text{ema}} \leftarrow \text{PowerIterationCompress}(G'_{\text{updated}}, r)$
    end for
end for
return $\mathcal{M}_{\theta_T}$
```

cosine similarity is negative. The projection is performed by projecting it onto the normal plane of $G^{l'}$: $\mathbf{g}^l_{\text{pc}} \leftarrow \mathbf{g}^l_{\text{pc}} - \frac{\mathbf{g}^l_{\text{pc}} \cdot G^{l'}}{\|G^{l'}\|^2} G^{l'}$.

The final consensus gradient, $\mathbf{g}_{\text{pc}}$, is the sum of these de-conflicted gradients from all languages: $\mathbf{g}_{\text{pc}} = \sum_{l \in L} \mathbf{g}^l_{\text{pc}}$. This vector represents a unified update direction that minimizes the interference.

Preference Filtering via Gradient In the subsequent training iteration, we use this consensus gradient $\mathbf{g}_{\text{pc}}$ as a reference for quality control. For each newly generated preference sample i in a language l, we compute its instantaneous gradient $\mathbf{g}^l_i$. We then measure its alignment with the consensus direction using cosine similarity, $\cos(\mathbf{g}^l_i, \mathbf{g}_{\text{pc}})$. By retaining only the samples with the highest similarity, we curate a dataset that promotes harmonious updates across languages, effectively filtering out those that would introduce conflict.

3.2 Efficient Gradient EMA via Incremental Low-Rank Updates

A practical challenge in our approach is the prohibitive memory cost of storing full EMA gradients for each language, especially for LLMs. To make our method feasible, we maintain a memory-efficient, low-rank approximation of the EMA gradient for each language, stored as a pair of factor matrices (P^l, Q^l).

Our method employs an incremental 'decompress-update-recompress' cycle to update these factors, ensuring the EMA calculation is mathematically sound while managing memory overhead. When a new mini-batch gradient $\mathbf{g}^l_\tau \in \mathbb{R}^{n \times m}$ arrives for language l, the update proceeds in three steps:

Decompress: First, we reconstruct the full-dimensional EMA gradient from the previous step using its stored factors: $G^l_{\text{ema},\tau-1} = P^l_{\text{ema},\tau-1}(Q^l_{\text{ema},\tau-1})^\top$.

Update: This recovered dense matrix is then updated with the new mini-batch gradient using the standard EMA formula in the full-dimensional space: $G'_{\text{updated}} = \gamma G^l_{\text{ema},\tau-1} + (1 - \gamma)\mathbf{g}^l_\tau$.

Re-compress: To restore memory efficiency, this newly updated, full-dimensional gradient G'_{updated} is immediately compressed back into a low-rank form. We use power iteration (Bathe and Wilson, 1972) to find its new rank-r factors, $P^l_{\text{ema},\tau}$ and $Q^l_{\text{ema},\tau}$, which replace the previous ones in storage.

Specifically, this process is performed for one matrix at a time. While the gradient for the active matrix is momentarily held in its dense form, the EMA gradients for all other matrices remain compressed, making the peak memory overhead manageable. After training, the final reconstructed gradient $G^l = P^l_{\text{ema}}(Q^l_{\text{ema}})^\top$ is then used for the consensus calculation.

3.3 Iterative Alignment with ConGrad

We integrate our ConGrad filtering method into an iterative self-rewarding procedure, as detailed in Algorithm 2. The training process begins at iteration $t = 1$ by fine-tuning the model on all self-generated preference data. For all subsequent iterations ($t > 1$), the ConGrad module is activated. It first computes the consensus gradient based on the previous iteration's training dynamics and then uses it to filter the newly generated preference dataset, ensuring that only high-quality, low-conflict samples are used for the model update.

Model optimization is performed using DPO. To mitigate the model's tendency towards verbosity, we incorporate a length penalty directly into the loss function (Park et al., 2024). The Length-Penalized DPO (LP-DPO) loss is defined as:

$$\mathcal{L}_{\text{LP-DPO}} = -\mathbb{E}_{(x,y_c,y_r)\sim\mathcal{D}}\left[\log\sigma\Big(\beta\, p_m + \alpha\, l_m\Big)\right], \quad (3)$$

where p_m is the standard preference margin between the chosen and rejected responses, and $l_m = |y_c| - |y_r|$ is the length margin. The hyperparameters β and α balance the preference-learning objective with the conciseness objective. By minimizing this loss, we guide the model to align with human preferences while maintaining conciseness.

4 Experimental Setup

We evaluate our self-rewarding multilingual LLM framework on several benchmarks and using different gradient filtering strategies. In particular, our experiments are designed to investigate the following three research questions:

- **RQ1**: Is iterative self-rewarding effective for multilingual LLM alignment?
- **RQ2**: Can our ConGrad filtering algorithm for preference data mitigate negative interference in multilingual LLM alignment?
- **RQ3**: How do different preference data filtering methods compare in terms of their effectiveness and characteristics?

Models We use widely adopted instruct version of `llama3-8b` and `gemma2-2b` as seed models for self-rewarding. Both models support multiple languages, but the capabilities are very uneven between languages. We chose two different sizes of models to see how the various algorithms perform in different parameter sizes.

Implementation Details In each round, for each prompt, we generate and score four responses and construct preference pairs. For preference data, **we filter the top 50% for each language** based on metrics and then perform one epoch of DPO training to ensure data quality and diversity. More details are in Appendix B.

4.1 Datasets and Metrics

Training Dataset is based on AlpaGasus (Chen et al., 2024a), which contains 9K high-quality English instruction following data filtered from the 52K Alpaca dataset (Taori et al., 2023). We randomly sample 1K prompts from AlpaGasus and use Google Translate to translate them into 9 languages: Italian (it), Chinese (zh), Portuguese (pt), Korean (ko), Spanish (es), German (de), Arabic (ar), Japanese (jp), and French (fr). For our multilingual experiments, we split the 1K prompts equally into 10 non-overlapping partitions, i.e., 100 prompts per language. For monolingual experiments, we use the full 1K prompts.

Evaluation Datasets include: (1) *aya evaluation suite* for instruction-following (Singh et al., 2024b). *aya evaluation suite* contains multilingual open-ended conversation-style prompts to evaluate multilingual open-ended generation quality. It is a high-quality prompt-based benchmark that contains translations and edits by human experts. (2) *Global MMLU* and multilingual version of *ARC challenge* for alignment tax (Singh et al., 2024a). *Global MMLU* improves upon previous translated MMLU variants by incorporating human-verified translations and annotating subsets for cultural sensitivity. This enables robust evaluation of LLMs across both culturally agnostic and culturally sensitive tasks. *ARC challenge* is a benchmark of grade-school science questions that are specifically selected to be unsolvable by simple retrieval or co-occurrence methods, requiring advanced reasoning and knowledge understanding from models (Clark et al., 2018; Lai et al., 2023).

Metrics include: (1) *Head-to-Head win rate* on *aya evaluation suite*. We utilize GPT-4o to compare the model after self-rewarding alignment during different iterations and the original base model, and calculate the win rate of the model after training over the seed model. (2) *5-shot Accuracy* on *Global MMLU* and multilingual version of *ARC challenge*.

Method/Round	Llama					Gemma				
	1	2	3	4	5	1	2	3	4	5
Mono	56.25	61.16	63.80	65.79	66.08	49.14	46.98	43.01	39.09	38.11
Mult-Full	57.57	60.94	61.75	64.42	66.52	57.80	57.93	57.31	57.51	59.53
UAB	57.57	61.51	63.04	64.51	66.54	57.80	58.73	59.63	60.61	60.85
LCL	57.57	60.48	63.96	65.76	66.67	57.80	59.65	60.04	60.52	62.51
Rand	57.57	61.56	64.06	65.21	66.32	57.80	56.27	57.98	54.25	57.25
Min-Len	57.57	58.19	59.13	61.55	63.57	57.80	51.56	58.94	52.66	54.29
Max-Len	57.57	60.33	59.88	61.11	67.03	57.80	58.83	58.84	58.94	59.52
Min-Reward	57.57	59.40	57.11	58.32	57.27	57.80	50.09	45.18	44.67	40.90
Max-Reward	57.57	64.66	**67.23**	**70.39**	70.64	57.80	**61.49**	63.73	67.72	61.82
Min-ConGrad	57.57	56.14	58.03	57.94	64.25	57.80	51.69	56.69	54.35	56.34
Max-ConGrad	57.57	**64.93**	66.58	69.99	**73.22**	57.80	60.31	**65.54**	**69.22**	**66.36**

Table 1: The average win rates of self-rewarding variants on aya evaluation suite. **Bold** indicates the best, underline the second-best.

4.2 Baselines

We compare our approach, ConGrad, against multiple baseline methods:

- **Monolingual Preference Alignment (Mono)** We perform self-rewarding using a full 1K prompt dataset for each language. This baseline serves as a reference point to multilingual baselines and provides evidence of negative interference if high-resource languages perform worse in the multilingual setting.

- **Multilingual Preference Alignment on Full Data (Mult-Full)** We perform self-rewarding across 10 languages, on the full preference dataset containing 100 prompts per language (1K total).

- **Multilingual Preference Alignment with Filtering Data** We compare our approach against several filtering techniques for DPO training: (i) Random filtering (**Rand**), where we report average performance of 3 runs. (ii) Length margin filtering, based on the length gap between chosen and rejected responses. We explore two variants: maximum length margin (**Max-Len**) and minimum length margin (**Min-Len**). (iii) Reward margin filtering based on the self-reward scores for the two responses. It can be viewed as the model's confidence in a sample, where a large margin indicates high confidence. We consider two variants: maximum reward margin (**Max-Reward**) and min reward margin (**Min-Reward**). (iv) Our **ConGrad** method, where we retain the top scoring samples based on the gradient similarity (**Max-ConGrad**). We also compare with a variant that retains the bottom-scoring samples (**Min-ConGrad**).

- **Task-Level Data Scheduling Strategies:** Uncertainty-Aware Balancing **(UAB)** uses uncertainty as the metric to adjust the sampler schedule in multilingual tasks. Based on the work of UAB, we use the self-reward preference margin as a proxy for model uncertainty in each language and adjust data sampling probabilities accordingly. Language Curriculum Learning **(LCL)** found that in multilingual tasks, when the amount of data is unbalanced, it is a good strategy to learn High-resource languages first and then all languages Choi et al. (2023). Inspired by LCL, we create a curriculum by first training on dominant languages (determined by Global MMLU scores) and then on all languages.

5 Experimental Results

5.1 Instruction Following Performance

Iterative Self-Rewarding Improves Multilingual Alignment As illustrated in Table 1, most variants of the self-rewarding procedure yield a sustained increase in win rates over successive iterations, demonstrating that the iterative self-rewarding strategy consistently improves instruction-following performance across different

Method/Round	Global MMLU (Seed Model: 48.67)					ARC Challenge (Seed Model: 72.23)				
	1	2	3	4	5	1	2	3	4	5
MULT-FULL	48.85	49.01	49.14	49.27	49.47	72.41	72.45	72.51	72.45	72.57
RAND	48.85	49.13	49.16	49.16	49.23	72.41	72.40	72.52	72.36	72.42
MIN-LEN	48.85	**50.40**	**50.47**	**50.52**	**50.62**	72.41	**72.82**	72.84	**72.99**	**72.99**
MAX-LEN	48.85	50.25	50.17	49.92	49.78	72.41	72.78	**72.85**	72.81	72.79
MIN-REWARD	48.85	49.03	49.19	49.40	49.40	72.41	72.40	72.54	72.60	72.66
MAX-REWARD	48.85	48.72	48.63	48.57	48.28	72.41	72.29	72.19	72.23	72.09
MIN-CONGRAD	48.85	49.00	49.07	49.15	49.11	72.41	72.40	72.49	72.32	72.41
MAX-CONGRAD	48.85	49.09	49.19	49.11	48.90	72.41	72.37	72.25	72.31	72.28

Table 2: The average accuracy of self-rewarding variations built on Llama. Red numbers indicate lower performance than the seed model. **Bold** indicates the best, underline the second-best.

Method/Round	Global MMLU (Seed Model: 38.88)					ARC Challenge (Seed Model: 62.85)				
	1	2	3	4	5	1	2	3	4	5
MULT-FULL	40.03	39.78	39.50	40.07	40.48	63.65	63.52	63.81	63.81	64.03
RAND	40.03	39.50	39.29	39.05	39.73	63.65	63.42	63.03	62.68	63.12
MIN-LEN	40.03	39.39	39.31	37.41	37.00	63.65	63.08	63.10	61.93	61.28
MAX-LEN	40.03	40.83	41.35	41.80	41.80	63.65	63.94	64.17	64.22	64.35
MIN-REWARD	40.03	39.78	38.68	38.32	36.40	63.65	63.41	63.04	62.91	61.99
MAX-REWARD	40.03	**40.87**	**41.93**	**41.97**	**42.48**	63.65	**63.96**	**64.35**	**64.45**	**64.48**
MIN-CONGRAD	40.03	40.29	39.54	40.98	38.81	63.65	63.56	63.29	63.61	62.81
MAX-CONGRAD	40.03	39.75	39.93	39.77	39.18	63.65	63.58	63.58	63.42	63.38

Table 3: The average accuracy of self-rewarding variations built on Gemma. Red numbers indicate lower performance than the seed model. **Bold** indicates the best, underline the second-best.

languages. In the Gemma experiments, our methods exhibited a decline in performance at the fifth round of iteration, suggesting that the benefits of self-rewarding may saturate or reverse due to overfitting.

CONGRAD Filtering Enhances Self-Rewarding and Outperforms Other Filtering Methods Compared to the vanilla self-rewarding algorithm MULT-FULL, CONGRAD further improves performance (73.22% vs 66.52% on Llama and 69.22% vs 59.53% on Gemma). This indicates two points: first, negative cross-lingual interference is present during multilingual alignment; second, our filtering method, which discards samples with low gradient similarity, can effectively mitigate this interference. While our method outperforms other filter methods, we also observe that MAX-REWARD is a strong baseline, because the base models have modest capacity, leading to noise during scoring. The method reduces noise in preference data by selecting samples with the highest confidence (i.e., MAX-REWARD). In contrast, filtering based on length differences only yields limited gains. Furthermore, task-level scheduling methods such as UAB and LCL only show limited improvement, which corresponds to a prevailing hypothesis that the quality of individual samples is often more crucial than the sheer quantity of data (Chen et al., 2024a; Zhou et al., 2023).

Coexistence of Positive Transfer and Negative Interference Across Languages Beyond negative interference, the results also indicate positive cross-lingual transfer. For instance, in the Gemma results, compared to the MONO, MULT-FULL significantly boosts performance for several languages. The average win rates on the best epoch increase from 49.14% to 59.53%, even though the data volume is controlled to be the same between the multilingual and monolingual settings. In more detailed results in Appendix E, we find that these improvements come mainly from underperforming languages such as Arabic, Korean, and German. This suggests that alignment improvements can largely stem from positive transfer from better-

Method/Round	Hindi 1	Hindi 2	Hindi 3	Hindi 4	Hindi 5	Vietnamese 1	Vietnamese 2	Vietnamese 3	Vietnamese 4	Vietnamese 5	Russian 1	Russian 2	Russian 3	Russian 4	Russian 5
Llama3-8B															
MULT-FULL	54.50	57.50	61.25	62.25	65	59.25	62.00	65.00	65.00	69.25	55.53	58.50	63.00	64.75	66.25
MAX-REWARD	54.50	**66.00**	67.25	69.25	71.00	59.25	63.00	68.50	70.75	**73.50**	55.53	**61.70**	**68.00**	**70.25**	71.25
MAX-CONGRAD	54.50	63.75	**68.00**	**70.25**	**73.75**	59.25	**63.75**	**71.75**	**71.25**	72.50	55.53	61.00	67.50	69.75	**71.50**
Gemma2-2B															
MULT-FULL	54.52	**59.55**	55.78	55.78	56.03	56.50	54.00	55.75	55.00	58.25	59.05	61.81	58.29	63.32	61.81
MAX-REWARD	54.52	54.52	61.56	63.07	57.54	56.50	60.00	62.50	63.00	**66.50**	59.05	63.32	**69.85**	65.83	64.32
MAX-CONGRAD	54.52	55.03	**62.81**	**63.82**	**58.04**	56.50	**62.00**	**64.00**	**71.75**	65.50	59.05	**66.08**	65.33	**69.60**	**67.09**

Table 4: The average win rates of self-rewarding variants on unseen languages from aya evaluation suite. **Bold** indicates the best, underline the second-best.

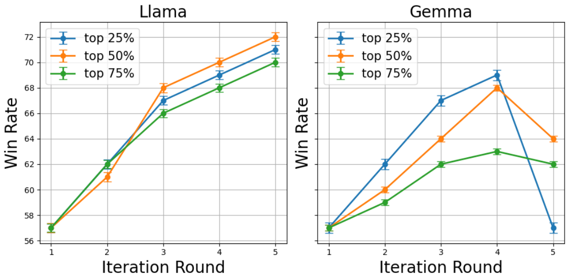

Figure 2: Win rate of different filter percentages.

Method	avg of others	lv	lt	sl	sk
MULT-FULL	67.7	69.5	74.25	74.25	68.0
REWARD	69.5	74.25	73.0	74.25	75.75
CONGRAD	**73.2**	**76.5**	**75.5**	**81.0**	**78.75**

Table 5: Win rates on underrepresented languages.

performing languages for languages with initially low performance.

Impact of Gradient Filter Percentage The percentage of the filter serves as an important hyperparameter that controls the strength of the filter algorithm. Ideally, the algorithm should not be too sensitive to the filtering strength within a certain range. In Figure 2, we report the win rate resulting from retaining different percentages of preference data while performing filtering. As shown, for Llama, retaining the top 50% of the data for each language yields the best performance but is overall insensitive to the filter strength. For Gemma, on the other hand, retaining 25% yields the best results at the best iteration round. This may be because the performance in Gemma is weaker compared to Llama, so there is more noise in the process of reward computation, and increasing the filter strength improves the performance to some extent.

5.2 Alignment Tax

Previous work has shown that after alignment training, such as DPO or RLHF, the model may forget some of its knowledge, which in turn leads to some degradation of the basic capabilities, also known as the alignment tax. Ideally, alignment algorithms should not produce much alignment tax. To test whether our algorithm suffers from alignment tax, we use two datasets for testing, *Global MMLU* and *ARC challenge*. For *Global MMLU*, we test the same ten languages. For *ARC challenge*, we test the other eight languages due to their lack of Korean and Japanese data. From the experimental results of Table 2 and Table 3, we can find that our algorithm even slightly improves the model's performance on both datasets without incurring an alignment tax. This may stem from the fact that the increase in the model's instruction following capability boosts its 5-shot in-context learning performance.

5.3 Analysis

Performance on Unseen Languages In some extreme cases, we may need to use certain languages that are not included in the self-rewarding process. To verify the usability of our model in such cases, we randomly selected three other widely varying languages, Hindi, Vietnamese and Russian, for validation. According to Table 4, multilingual self-rewarding generalizes well to unseen languages, indicating a positive transfer between languages. In addition, better results can still be achieved using CONGRAD, illustrating the generalizability of our approach.

Gains in Underrepresented Languages Current LLMs often lack capability in many underrepresented languages. To test the robustness of CONGRAD in these settings, we added four new languages with minimal `llama3-8b` tokenizer support (around 800+ tokens), Latvian (lv), Lithuanian (lt), Slovenian (sl), and Slovak (sk), to the original ten languages for alignment. The results,

Conflict Type	Example (Prompt & Responses)	Analysis of Negative Interference
Case 1: Cultural Divergence *(Source: Chinese)*	**Prompt:** Describe a special occasion dinner. **Chosen Response:** Describes a traditional Chinese family reunion with "dumplings" and "grandmother's secret recipe." **Rejected Response:** Describes a Western-fantasy banquet with "roast beef" and "gem-encrusted walls."	The gradient update to align with this specific Chinese cultural preference points in a significantly different direction than the cross-lingual consensus. While this is valid data, naive joint training struggles to reconcile these deep cultural divergences, leading to gradient conflicts that ConGrad successfully identifies.
Case 2: Linguistic Untranslatability *(Source: English)*	**Prompt:** Create a funny slogan for a new ice cream shop. **Chosen Response:** Relies on English puns like "The cream of the crop" or "Melt your heart."	Puns rely on language-specific phonology rather than shared semantics. Gradients optimizing for English wordplay may act as noise for other languages (e.g., Chinese/German) where these puns do not transfer. ConGrad flags this as "high-conflict" because it diverges from the shared semantic alignment objective.

Table 6: **Qualitative Analysis of Filtered Samples.** Examples of samples identified by ConGrad as having high negative interference (low cosine similarity with consensus gradient). ConGrad filters out samples causing cultural misalignment or linguistic untranslatability to focus on cross-lingually sharable knowledge.

Operation	Without Filtering	ConGrad	Delta
Per Epoch (Based on Gemma-2-2B)			
1. Data Generation	376s	376s	-
2. Annotation (Judge)	170s	170s	-
3. **Filtering**	**0s**	**70s**	**+70s**
4. DPO Training	169s	191s	+22s
Total Time (5 Epochs)	**∼62 min**	**∼69 min**	**∼1.1x**

Table 7: **Runtime Efficiency Analysis.** Comparison of training time between Naive Self-Rewarding and ConGrad. The gradient-based filtering and EMA decompression/recompression adds marginal overhead (+70s/epoch and +22s/epoch) compared to the dominant inference phases. Total overhead is approximately 1.1×, demonstrating scalability.

summarized in Table 5, show that CONGRAD improves alignment in these languages while maintaining high performance of others. CONGRAD consistently outperformed the baselines in all four underrepresented languages and on the original ten. These findings suggest CONGRAD enhances alignment even with limited base model support, indicating its generalization potential.

Runtime Analysis We provide a rigorous runtime analysis in Table 7. While ConGrad introduces a gradient computation step during filtering, this phase requires no optimizer state maintenance. With our low-rank compression, the memory usage for storing gradients of 10 languages (Gemma-2-2B, projection dimension=64) is only 3.3GB per GPU. The total training time overhead is approximately 1.1× compared to the naive baseline, as the pipeline is dominated by data generation and annotation, making the alignment tax of ConGrad negligible in practice.

Qualitative Analysis of Filtered Samples To provide concrete insight into the nature of "negative interference," we qualitatively analyzed the samples identified by ConGrad as having the lowest gradient similarity to the cross-lingual consensus. As illustrated in **Table 6**, these conflicts primarily stem from two sources: *Cultural Divergence* and *Linguistic Untranslatability*. For instance, in Case 1, a Chinese-specific description of a "special dinner" (focusing on dumplings and kinship) presents an optimization direction that opposes the Western-centric consensus often dominated by English data. Similarly, in Case 2, language-specific puns act as noise for cross-lingual alignment because they rely on phonology rather than shared semantics. By filtering these high-conflict samples, ConGrad allows the model to prioritize cross-lingually transferable knowledge, thereby mitigating interference.

6 Conclusion

We propose CONGRAD, a multilingual LLM alignment framework with conflicting gradient filtering, which avoids the reliance on external annotations. Our approach iteratively generates and scores preference data and introduces a PCGrad-based gradient filtering strategy to mitigate negative cross-lingual interference. Extensive experiments on LLaMA3 and Gemma2 show that our approach significantly improves instruction following for underrepresented languages, maintains performance for mainstream languages, and generalizes to unseen languages. Furthermore, our analysis highlights the dual role of positive transfer and negative interference in multilingual training, suggesting that careful data selection is essential to fully unlock the multilingual potential of aligned LLMs.

Limitations

While CONGRAD demonstrates strong empirical performance in mitigating negative cross-lingual interference, several limitations remain. First, although sublinear gradient compression significantly reduces memory overhead, it introduces approximation noise that may impact the accuracy of gradient filtering. Second, our experiments are limited to typologically diverse but relatively balanced languages; generalization to highly imbalanced or severely low-resource language scenarios remains an open challenge. However, this may be more of a problem with the pre-training stage than with the alignment stage since the LLM's most basic language capabilities still require the support of pre-trained data. Lastly, extending the framework to support dynamic filtering during training, rather than round-level static selection, could enable finer-grained control over training dynamics but might also introduce additional costs.

Future work can explore hybrid filtering strategies that combine gradient-based and reward-based signals to improve sample selection. Additionally, integrating cross-lingual alignment objectives directly into the optimization process, beyond data filtering, may further enhance multilingual alignment.

Acknowledgement

This research was supported by an eBay Research Award and Monash eResearch capabilities, including M3. The authors are grateful to the anonymous reviewers for their helpful comments.

References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, and 1 others. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and 1 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Klaus-Jürgen Bathe and Edward L Wilson. 1972. Large eigenvalue problems in dynamic analysis. *Journal of the Engineering Mechanics Division*, 98(6):1471–1485.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. 2024a. Alpagasus: Training a better alpaca with fewer data. In *The Twelfth International Conference on Learning Representations*.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and 1 others. 2024b. Alpagasus: Training a better alpaca with fewer data. In *The Twelfth International Conference on Learning Representations*.

Pinzhen Chen, Shaoxiong Ji, Nikolay Bogoychev, Andrey Kutuzov, Barry Haddow, and Kenneth Heafield. 2024c. Monolingual or multilingual instruction tuning: Which makes a better alpaca. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1347–1356, St. Julian's, Malta. Association for Computational Linguistics.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. 2024d. Self-play fine-tuning convertsweak language models to strong language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Dami Choi, Derrick Xin, Hamid Dadkhahi, Justin Gilmer, Ankush Garg, Orhan Firat, Chih-Kuan Yeh, Andrew M Dai, and Behrooz Ghorbani. 2023. Order matters in the presence of dataset imbalance for multilingual learning. *Advances in Neural Information Processing Systems*, 36:66902–66922.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024. Ultrafeedback: boosting language models with scaled ai feedback. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

John Dang, Arash Ahmadian, Kelly Marchisio, Julia Kreutzer, Ahmet Üstün, and Sara Hooker. 2024. RLHF can speak many languages: Unlocking multilingual preference optimization for LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13134–

13156, Miami, Florida, USA. Association for Computational Linguistics.

Yue Deng, Wenxuan Zhang, Sinno Jialin Pan, and Lidong Bing. 2023. Multilingual jailbreak challenges in large language models. *arXiv preprint arXiv:2310.06474*.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. 2023. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36:30039–30069.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Srishti Gureja, Lester James V Miranda, Shayekh Bin Islam, Rishabh Maheshwary, Drishti Sharma, Gusti Winata, Nathan Lambert, Sebastian Ruder, Sara Hooker, and Marzieh Fadaee. 2024. M-rewardbench: Evaluating reward models in multilingual settings. *arXiv preprint arXiv:2410.15522*.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Jian Hu, Xibin Wu, Zilin Zhu, Weixun Wang, Dehao Zhang, Yu Cao, and 1 others. 2024. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework. *arXiv preprint arXiv:2405.11143*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Md Tawkat Islam Khondaker, Abdul Waheed, El Moatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2023. GPTAraEval: A comprehensive evaluation of ChatGPT on Arabic NLP. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 220–247, Singapore. Association for Computational Linguistics.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. 2024. Understanding the effects of RLHF on LLM generalisation and diversity. In *The Twelfth International Conference on Learning Representations*.

Hadas Kotek, Rikker Dockum, and David Sun. 2023. Gender bias and stereotypes in large language models. In *Proceedings of The ACM Collective Intelligence Conference*, CI '23, page 12–24, New York, NY, USA. Association for Computing Machinery.

Vitaly Kurin, Alessandro De Palma, Ilya Kostrikov, Shimon Whiteson, and Pawan K Mudigonda. 2022. In defense of the unitary scalarization for deep multitask learning. *Advances in Neural Information Processing Systems*, 35:12169–12183.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626.

Viet Lai, Chien Nguyen, Nghia Ngo, Thuat Nguyen, Franck Dernoncourt, Ryan Rossi, and Thien Nguyen. 2023. Okapi: Instruction-tuned large language models in multiple languages with reinforcement learning from human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 318–327, Singapore. Association for Computational Linguistics.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. 2024. Rlaif vs. rlhf: scaling reinforcement learning from human feedback with ai feedback. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Chong Li, Shaonan Wang, Jiajun Zhang, and Chengqing Zong. 2024. Improving in-context learning of multilingual generative language models with cross-lingual alignment. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8058–8076, Mexico City, Mexico. Association for Computational Linguistics.

Yong Lin, Hangyu Lin, Wei Xiong, Shizhe Diao, Jianmeng Liu, Jipeng Zhang, Rui Pan, Haoxiang Wang, Wenbin Hu, Hanning Zhang, Hanze Dong, Renjie Pi, Han Zhao, Nan Jiang, Heng Ji, Yuan Yao, and Tong Zhang. 2024. Mitigating the alignment tax of RLHF. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 580–606, Miami, Florida, USA. Association for Computational Linguistics.

Itai Mondshine, Tzuf Paz-Argaman, and Reut Tsarfaty. 2025. Beyond english: The impact of prompt translation strategies across languages and tasks in multilingual llms. *arXiv preprint arXiv:2502.09331*.

Tetsuro Morimura, Mitsuki Sakamoto, Yuu Jinnai, Kenshi Abe, and Kaito Ariu. 2024. Filtered direct preference optimization. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22729–22770.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Richard Yuanzhe Pang, Weizhe Yuan, He He, Kyunghyun Cho, Sainbayar Sukhbaatar, and Jason E Weston. 2024. Iterative reasoning preference optimization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. 2024. Disentangling length from quality in direct preference optimization. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 4998–5017.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Reva Schwartz, Reva Schwartz, Apostol Vassilev, Kristen Greene, Lori Perine, Andrew Burt, and Patrick Hall. 2022. *Towards a standard for identifying and managing bias in artificial intelligence*, volume 3. US Department of Commerce, National Institute of Standards and Technology ....

Ozan Sener and Vladlen Koltun. 2018. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31.

Uri Shaham, Jonathan Herzig, Roee Aharoni, Idan Szpektor, Reut Tsarfaty, and Matan Eyal. 2024. Multilingual instruction tuning with just a pinch of multilinguality. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 2304–2317.

Shivalika Singh, Angelika Romanou, Clémentine Fourrier, David I Adelani, Jian Gang Ngui, Daniel Vila-Suero, Peerat Limkonchotiwat, Kelly Marchisio, Wei Qi Leong, Yosephine Susanto, and 1 others. 2024a. Global mmlu: Understanding and addressing cultural and linguistic biases in multilingual evaluation. *arXiv preprint arXiv:2412.03304*.

Shivalika Singh, Freddie Vargus, Daniel D'souza, Börje Karlsson, Abinaya Mahendiran, Wei-Yin Ko, Herumb Shandilya, Jay Patel, Deividas Mataciunas, Laura O'Mahony, Mike Zhang, Ramith Hettiarachchi, Joseph Wilson, Marina Machado, Luisa Moura, Dominik Krzemiński, Hakimeh Fadaei, Irem Ergun, Ifeoma Okoh, and 14 others. 2024b. Aya dataset: An open-access collection for multilingual instruction tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11521–11567, Bangkok, Thailand. Association for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Yuxuan Tong, Xiwen Zhang, Rui Wang, Ruidong Wu, and Junxian He. 2024. Dart-math: Difficulty-aware rejection tuning for mathematical problem-solving. *Advances in Neural Information Processing Systems*, 37:7821–7846.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Aniket Vashishtha, Kabir Ahuja, and Sunayana Sitaram. 2023. On evaluating and mitigating gender biases in multilingual settings. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 307–318, Toronto, Canada. Association for Computational Linguistics.

Zirui Wang, Zachary C. Lipton, and Yulia Tsvetkov. 2020. On negative interference in multilingual models: Findings and a meta-learning treatment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4438–4450, Online. Association for Computational Linguistics.

Zirui Wang, Yulia Tsvetkov, Orhan Firat, and Yuan Cao. 2021. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. In *International Conference on Learning Representations*.

Minghao Wu, Yitong Li, Meng Zhang, Liangyou Li, Gholamreza Haffari, and Qun Liu. 2021. Uncertainty-aware balancing for multilingual and multi-domain neural machine translation training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7291–7305, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Minghao Wu, Thuy-Trang Vu, Lizhen Qu, and Reza Haf. 2024a. Mixture-of-skills: Learning to optimize data usage for fine-tuning large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14226–14240, Miami, Florida, USA. Association for Computational Linguistics.

Minghao Wu, Thuy-Trang Vu, Lizhen Qu, and Gholamreza Haffari. 2024b. The best of both worlds: Bridging quality and diversity in data selection with bipartite graph. *arXiv preprint arXiv:2410.12458*.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. LESS: Selecting influential data for targeted instruction tuning. In *International Conference on Machine Learning (ICML)*.

Wen Yang, Junhong Wu, Chen Wang, Chengqing Zong, and Jiajun Zhang. 2025a. Implicit cross-lingual rewarding for efficient multilingual preference alignment. *arXiv preprint arXiv:2503.04647*.

Wen Yang, Junhong Wu, Chen Wang, Chengqing Zong, and Jiajun Zhang. 2025b. Language imbalance driven rewarding for multilingual self-improving. In *The Thirteenth International Conference on Learning Representations*.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33:5824–5836.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Yuanchi Zhang, Yile Wang, Zijun Liu, Shuo Wang, Xiaolong Wang, Peng Li, Maosong Sun, and Yang Liu. 2024. Enhancing multilingual capabilities of large language models through self-distillation from resource-rich languages. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11189–11204, Bangkok, Thailand. Association for Computational Linguistics.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, and 1 others. 2023. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36:55006–55021.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A Related Work

Multilingual Preference Optimization Recent work has explored bridging the performance gap between high- and low-resource languages through multilingual preference optimization. Li et al. (2024); Zhang et al. (2024) align internal representations and outputs using contrastive learning and instruction tuning. Yang et al. (2025b) leverage language imbalance as a natural preference signal for iterative self-improvement across languages, while Yang et al. (2025a) transfer implicit rewards from English-aligned models to other languages without explicit multilingual preference data. Dang et al. (2024) show that cross-lingual transfer emerges from multilingual preference training, with online methods outperforming offline ones. Zhang et al. (2024) propose a self-distillation method to improve multilingual generation by utilizing strong responses in resource-rich languages. Meanwhile, Gureja et al. (2024) reveal significant performance gaps in multilingual reward models, emphasizing the need for more robust evaluation and alignment across languages.

While our method draws inspiration from optimization-centric approaches such as PCGrad (Yu et al., 2020; Wang et al., 2021) for conflict resolution, applying PCGrad directly at the update step is computationally prohibitive for LLMs. Standard PCGrad requires storing full gradient vectors for each language and performing multiple backward passes per iteration, causing memory usage to scale linearly with the number of languages. ConGrad decouples this process: it moves conflict resolution to the data selection phase (offline filtering). This avoids the runtime memory bottleneck, making gradient-based alignment analysis feasible for 8B+ parameter models.

Synthetic Multilingual Dataset Creation Many studies create multilingual preference data via translation from English instructions (Lai et al., 2023; Shaham et al., 2024; Mondshine et al., 2025). Okapi (Lai et al., 2023) translates English prompts into 26 languages and ranks responses using GPT-3.5. Shaham et al. (2024) find that adding just a few multilingual examples improves cross-lingual generalization. Mondshine et al. (2025) propose selective pre-translation of prompt components, improving performance across 35 languages. Other work leverages language imbalance as a heuristic to generate multilingual preference pairs (Yang et al., 2025b).

Instruction and Preference Data Filtering Instruction data filtering methods aim to identify the most useful samples for fine-tuning. LESS (Xia et al., 2024) selects influential instructions by estimating gradient similarity with few-shot targets, while DART-Math (Tong et al., 2024) prioritizes difficult queries during data synthesis to enhance reasoning. Alpagasus (Chen et al., 2024b) filters low-quality instruction-response pairs using GPT-based scoring. Wu et al. (2024b) balance quality and diversity via a graph-based selector. For preference data, fDPO (Morimura et al., 2024) introduces reward-model-based filtering during DPO training, showing that preference data quality critically impacts alignment. Overall, most prior work focuses on instruction data, while systematic filtering of preference data remains underexplored.

Theoretical Grounding and Connection to MTL. Although heuristic in implementation, ConGrad is grounded in the theory of Pareto Optimality in Multi-Task Learning (MTL) (Sener and Koltun, 2018). In multilingual alignment, we treat each language as an independent task. Gradient conflicts indicate a non-Pareto stable state where improving one language harms another. By filtering based on a consensus gradient, ConGrad seeks a descent direction that improves (or maintains) all task objectives, promoting cross-lingual fairness.

B Implementation Details

To optimize model performance, we conducted systematic hyperparameter tuning during training. We tuned key hyperparameters, including the learning rate, compression dimension, and the length penalty strength α used in DPO training. Specifically, we searched the learning rate over $[2 \times 10^{-6}, 1 \times 10^{-6}, 5 \times 10^{-7}]$, while the batch size was fixed at 16 and the compression rank was fixed at 64 to balance GPU memory constraints and training stability, which is similar with LoRA (Hu et al.). When performing power iteration, we set the number of iterations to 3 to ensure a balance between precision and efficiency. The length penalty strength parameter was explored within [0.02, 0.01, 0.005].

For LLama, all training was performed on 4 A100 GPUs, and for Gemma, all training was performed on 2 A100 GPUs. We use the openrlhf framework (Hu et al., 2024) for DPO training and vLLM (Kwon et al., 2023) for inference.

C Further Analysis

C.1 Gradient Approximation Quality and Sensitivity Analysis

Approximation Quality We first evaluated the approximation error of our gradient compression technique. For the chosen rank of $r = 64$, the recovered low-rank gradients exhibited a strong directional correlation with the original full-rank gradients, achieving an average cosine similarity of approximately 0.8. We determined that this level of directional approximation was adequate to guide the data filtering process effectively without requiring the full, computationally expensive gradients.

Sensitivity to Compression Ratio To investigate the model's sensitivity to the degree of compression, we conducted an ablation study on the `Llama3-8B` model with multiple varied ranks. The results, summarized in Table 8, demonstrate that the performance of CONGRAD is robust to the choice of rank within a reasonable range ($r \geq 8$), showing only marginal fluctuations. A significant performance degradation was observed only when the compression became highly aggressive ($r = 4$). We attribute this robustness to our data selection mechanism, which filters the top 50% of data based on gradient scores. This approach is likely resilient to minor inaccuracies in gradient approximation, as the relative ranking of the most beneficial samples is largely preserved even with compressed gradients.

Table 8: Performance sensitivity of CONGRAD on Llama3-8B with varying compression ranks (r).

Rank (r)	**Average Win Rate** (%)
64	73.2
32	72.6
16	72.3
8	71.9
4	67.5

D Self-rewarding Iterative DPO Prompts

Response Evaluation Prompt for Self-Rewarding

`Review the user's question and the corresponding response using the **additive 5-point scoring system** described below. Points are accumulated based on the satisfaction of each criterion:`

`- Add 1 point (total Score: 1) if the response is relevant and provides some information related to the user's inquiry, even if it is incomplete or contains some irrelevant content.`

`- Add another point (total Score: 2) if the response addresses a substantial portion of the user's question, but does not completely resolve the query or provide a direct answer.`

`- Add another point (total Score: 3) if the response answers the basic elements of the user's question in a useful way, regardless of whether it seems to have been written by an AI Assistant or if it has elements typically found in blogs or search results.`

`- Add another point (total Score: 4) if the response is clearly written from an AI Assistant's perspective, addressing the user's question directly and comprehensively, and is well-organized and helpful, even if there is slight room for improvement in clarity, conciseness or focus.`

`- Add another point (total Score: 5) for a response that is impeccably tailored to the user's question by an AI Assistant, without extraneous information, reflecting expert knowledge, and demonstrating a high-quality, engaging, and insightful answer.`

`User: 0`

```
<response>1</response>
Remember to assess from the
AI Assistant perspective.
To evaluate the response in
alignment with this additive
scoring model, we’ll
systematically attribute
points based on the outlined
criteria.
After examining the user’s
instruction and the
response:
- Briefly analyse the
response in *English*, *up
to 100 words*, which is a
*strict limit*, from the AI
Assistant perspective.
- ** Conclude with the score
of the response *strictly
using English* and the
format: “Score: <total
points>” **
- The score should be an
**integer from 0 to 5**
```

E Addition Results

	it	zh	pt	en	ko	es	de	ar	ja	fr	avg
Mono	52.50	71.20	59.20	55.75	76.00	60.64	64.00	71.26	76.50	73.76	66.08
Mult-Full	61.75	64.60	66.00	54.56	75.00	67.82	63.00	63.98	74.00	74.50	66.52
UAB	62.50	71.29	64.40	53.00	72.75	70.00	60.75	59.00	74.75	77.00	66.54
LCL	67.00	72.09	63.40	54.00	77.25	67.75	62.50	59.00	76.25	77.50	67.67
Rand	63.50	70.88	63.20	53.37	74.75	66.83	62.25	59.72	72.75	75.99	66.32
Min-Len	58.00	69.00	65.40	53.17	70.50	58.17	58.75	64.17	71.00	67.57	63.57
Max-Len	63.25	65.86	64.20	53.17	75.50	66.83	63.00	69.49	77.00	72.03	67.03
Min-Reward	55.75	59.44	56.20	45.63	60.00	58.25	58.75	55.91	59.25	63.61	57.27
Max-Reward	69.75	**76.91**	70.40	**58.93**	78.25	71.50	66.25	59.45	76.50	78.47	70.64
Min-ConGrad	69.00	65.26	62.80	60.12	65.00	61.31	63.50	61.00	67.00	67.50	64.25
Max-ConGrad	**71.00**	74.50	**75.40**	57.14	**82.25**	**72.75**	**71.50**	**65.16**	**82.00**	**80.50**	**73.22**

Table 9: The win rate of self-rewarding variations of LLama3-8B evaluated on aya evaluation suite for each language in the final round. The best results are highlighted in **bold**, and the second-best results are highlighted in underline.

	it	zh	pt	en	ko	es	de	ar	ja	fr	avg
Mono	43.50	62.65	49.60	55.40	45.25	50.25	46.50	31.22	46.75	60.30	49.14
mult-full	61.00	62.45	59.00	58.40	70.00	51.25	56.75	56.85	60.50	59.05	59.53
UAB	67.50	59.84	58.40	59.60	70.00	51.75	56.50	59.40	65.00	60.50	60.85
LCL	68.50	62.24	60.80	57.60	74.00	55.25	59.00	61.20	65.75	60.75	62.51
Rand	58.25	63.45	58.20	55.40	63.00	54.27	56.25	60.00	54.25	56.75	57.98
Min-Len	51.01	57.29	53.20	52.78	61.75	42.50	45.00	58.10	51.75	42.25	51.56
Max-Len	61.75	63.45	58.40	58.73	63.25	51.76	59.50	60.84	59.25	58.25	59.52
Min-Reward	50.00	50.00	46.80	47.40	55.00	44.72	52.75	59.64	47.50	49.75	50.09
Max-Reward	**70.75**	69.68	63.80	61.51	72.25	**69.60**	64.75	**67.07**	69.25	68.59	67.72
Min-ConGrad	63.25	57.03	54.00	52.38	63.00	49.75	56.00	56.20	52.75	59.05	56.34
Max-ConGrad	68.00	**72.29**	**66.40**	**63.49**	**76.25**	69.50	**66.25**	65.35	**76.00**	**68.75**	**69.22**

Table 10: The win rate of self-rewarding variations of Gemma2-2B evaluated on aya evaluation suite for each language in the fourth round. The best results are highlighted in **bold**, and the second-best results are highlighted in underline.