AIF-GEN: OPEN-SOURCE PLATFORM AND SYNTHETIC DATASET SUITE FOR LIFELONG REINFORCEMENT LEARNING ON LARGE LANGUAGE MODELS

Shahrad Mohammadzadeh[*,1,3,‡] **Jacob Chmura**[*,1,3] **Ivan Anokhin**[†,2,3]
Jacob-Junqi Tian[†,4] **Mandana Samiei**[†,1,3] **Taz Scott-Talib**[1]
Irina Rish[2,3,5] **Doina Precup**[1,3,5] **Reihaneh Rabbany**[1,3,5] **Nishanth Anand**[1,3]

ABSTRACT

Reinforcement learning has proven effective for fine-tuning large language models (LLMs) using reward models trained on human preference data. However, collecting such feedback remains expensive, especially in dynamic settings like personalized tutoring, where users' preferences shift over time and through past interactions. These non-stationarities pose challenges for studying lifelong learning in RLHF pipelines, a growing concern as LLMs are increasingly deployed in real-world systems that demand continual adaptation. To address this, we present AIF-GEN, the first synthetic preference data generation platform designed for traditional and lifelong RLHF. We use AIF-GEN to instantiate 18 synthetic datasets grouped into 4 non-stationary meta-datasets. Through experiments on various synthetic benchmarks, we find that RL algorithms must be tailored to the specific type of non-stationarity they encounter. Our results show AIF-GEN's potential to support the development of RLHF algorithms that continually align LLMs.

Code: AIF-GEN

Data & Dataset Cards: https://huggingface.co/LifelongAlignment

Documentation: aif-gen.readthedocs.io

1 INTRODUCTION

Reinforcement learning from human feedback (RLHF) has emerged as a critical technique for aligning large language models (LLMs) with human intentions (Christiano et al., 2017; Stiennon et al., 2020; Ouyang et al., 2022; Bai et al., 2022), particularly in tasks requiring nuanced judgments such as helpfulness, factual accuracy (Sun et al., 2023), and safety (Dai et al., 2023). Despite its effectiveness, RLHF relies on costly, static human data, limiting adaptability in dynamic settings like personalized education, where user preferences evolve over time (Jurenka et al., 2024).

To address these limitations, synthetic data generation methods offer scalable alternatives by using LLM-generated annotations to reduce the cost and complexity of human preference collection, which is then used to align LLMs— an approach called reinforcement learning from AI feedback (RLAIF) (Li & Chen, 2023; Zhang et al., 2023). However, these methods generate static dataset for one-time alignment process. This limitation hinders the progress in lifelong alignment, where models must adapt to distributional drifts. Without robust frameworks for generating preference dataset that reflect real-world variability, LLMs are prone to catastrophic forgetting and overfitting to narrow feedback signals (Parisi et al., 2019), thereby limiting their applicability in real-world applications.

In this work, we introduce AIF-GEN—the first platform for scalable synthetic data generation under evolving preferences. AIF-GEN allows systematic generation of non-stationary datasets by parameterizing evolving user objectives, domains, and preferences, thereby facilitating progress in fine-tuning and continual alignment of LLMs. Using the platform, we instantiate 18 synthetic preference datasets, grouped into 4 meta-datasets with evolving objectives, totalling roughly 170,000

[*]Equal contribution; [†]Equal contribution; [1]McGill University; [2]Université de Montréal; [3]Mila - Quebec AI Institute; [4]Vector Institute; [5]CIFAR AI Chair; [‡]Correspondence to: shahrad.mohammadzadeh@mila.quebec

prompts and 340,000 preference annotations. We illustrate that the performance of standard RL algorithms (e.g., PPO (Schulman et al., 2017), DPO (Rafailov et al., 2024)) varies significantly under different types of distributional drift. Designed to be LLM-agnostic, scalable, and customizable, AIF-GEN lowers the barrier to entry for lifelong RLHF research and provides a foundation for advancing adaptive, preference-aligned language models.

Summary of contributions:

1. We introduce lifelong RLHF and categorize types of non-stationarity in RLHF (§ 4);
2. We present **AIF-GEN**, the first synthetic data generation platform tailored to lifelong RLHF (§ 5);
3. Using AIF-GEN, we generate a diverse suite of synthetic datasets that capture varying types and degrees of non-stationarity to support controlled experiments(§ 6.1);
4. We validate the quality of our synthetic datasets using standard metrics from the natural language processing literature and human evaluation(§ 6.2);
5. Through experiments with PPO (Schulman et al., 2017), DPO (Rafailov et al., 2024), and CPPO (Zhang et al., 2024b), we show performance varies by non-stationarity type, highlighting the need for further research in lifelong RLHF methods (§ 7).

2 Related Work

RLHF leverages human feedback to fine-tune LLMs by optimizing a learned reward function that encapsulates human preferences (Christiano et al., 2017; Ouyang et al., 2022). Variants of this framework, illustrated in Figure 1, include *single-shot RLHF*, which involves a one-time fine-tuning process; *iterative RLHF*, where the models are repeatedly fine-tuned through multiple rounds of feedback (Stiennon et al., 2022; Ouyang et al., 2022); *lifelong RLHF*, where the models are trained continually to adapt to the evolving user preferences and non-stationarity tasks (Zhang et al., 2024b;a).

RLHF algorithms commonly leverage high-quality human preference datasets such as *HH-RLHF* (Kaplan et al., 2022) to align LLMs. These datasets rely solely on human annotations, making the data-curation process costly and difficult to scale. Large organizations rely on carefully curated private datasets for alignment, which makes it more challenging for independent researchers to work on RLHF. Open-source efforts like *Open Assistant* have improved accessibility, but replicating such large-scale human annotation pipelines remains financially and logistically challenging.

To overcome the limitations posed by human annotations, synthetic data generation is seen as an effective alternative (Li & Chen, 2023). Recent large-scale synthetic datasets like *Ultra-Feedback* (Cui et al., 2023) highlight the promise of model-generated preference data for LLM alignment but lack support for non-stationarity and are not open-sourced. Similarly, frameworks such as *DataDreamer* (Patel et al., 2024) and *Curator* (Marten et al., 2025) offer flexible data curation tools but do not explicitly address evolving preferences or provide quality validation for generated data.

AIF-GEN, our platform, addresses the above-mentioned gaps by combining the strengths of prior frameworks—open-source accessibility, synthetic data flexibility, and validation—with support for non-stationary preferences/prompts for lifelong RLHF research. A detailed breakdown is shown in Table 1.

Library/Feature	HH-RLHF	Ultra-Feedback	OpenAssistant	DataDreamer	Curator	AIF-GEN (ours)
Open Source	×	×	✓	✓	✓	✓
Non-Stationarity Support	×	×	×	×	×	✓
Validation Metrics	✓	✓	×	×	×	✓
Human Verified	✓	✓	✓	×	×	✓
Caching	×	×	✓	✓	✓	✓
HuggingFace Compatible	✓	✓	✓	✓	✓	✓
Customizable Dataset	×	×	×	✓	✓	✓

Table 1: AIF-GEN is the first open source synthetic data generation tool offering full prompt and preference customization with native support for non-stationarity and evolving preferences.

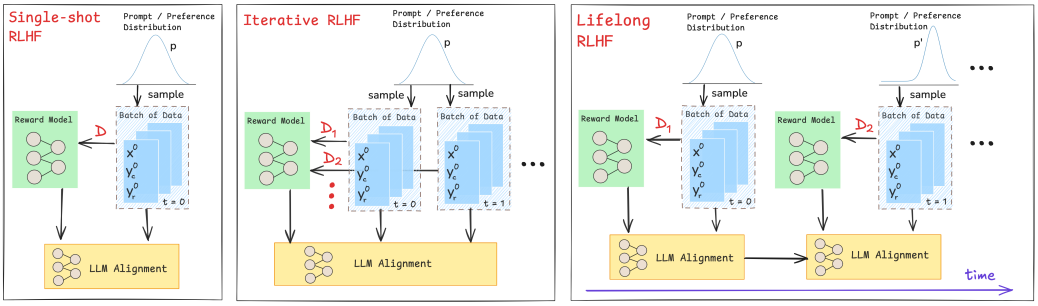


Figure 1: Illustration of RLHF variants. **Left:** Single-shot RLHF aligns the LLM using a fixed reward model trained once on static preferences. **Middle:** Iterative RLHF repeatedly updates the reward model and LLM alignment in multiple iterations using batches of data sampled from a fixed distribution (Xiong et al., 2024). **Right:** Lifelong RLHF extends this paradigm by continually adapting to shifting preference distributions, enabling the LLM to dynamically realign to evolving human preferences.

3 Background

The RLHF process begins after a base LLM, π^0, is supervised fine-tuned on a specific dataset to obtain an SFT model, π^{SFT}. This model is queried with prompts, x, to get two responses, (y_1, y_2). These responses are then passed on to humans (or an LLM judge in the case of RLAIF) to select a *preferred or chosen* response; the other response becomes the *rejected* response. For example, if y_1 was the chosen response, then y_2 becomes the rejected response, and we say that y_1 is preferred over y_2 for prompt x, $y_1 \succ y_2|x$. The preferences are assumed to come from an unobservable reward function, $r^*(y, x)$, which is usually modelled as the Bradley-Terry model (Bradley & Terry, 1952) due to its simplicity:

$$p^*(y_1 \succ y_2|x) = \frac{\exp\left(r^*(y_1, x)\right)}{\exp\left(r^*(y_1, x)\right) + \exp\left(r^*(y_2, x)\right)}. \tag{1}$$

Other options for reward modelling, such as Nash (Munos et al., 2024; Zhu et al., 2024) and Plackett-Luce (Plackett, 1975), are also explored in the literature.

Typically, several pairs of responses are generated for a set of prompts to form a preference dataset, $D = \{x^i, y_c^i, y_r^i\}_{i=1}^N$, where y_c^i is the chosen response and y_r^i is the rejected response for the prompt x^i. Then, a reward model is learned using this dataset, which provides the training signal for RL algorithms to fine-tune the LLM. Here, the prompt distribution from which the prompts, x, are sampled and the underlying reward function, $r^*(y, x)$, are assumed to be stationary. In the RL fine-tuning step, the SFT model, π^{SFT}, is optimized to generate responses that maximize rewards coming from the learned reward model:

$$max_{\pi_\theta} \mathbb{E}_{x\sim D, y\sim\pi_\theta(\cdot|x)}[r_\phi(y, x) - \beta\mathbb{D}_{KL}(\pi_\theta(\cdot|x)||\pi^{SFT}(\cdot|x))], \tag{2}$$

where r_ϕ is the learned reward model parameterized by ϕ, β is a regularization parameter to control the deviation from the SFT model, π^{SFT}, to preserve the safety and other alignment operations. Although any RL algorithm can optimize the objective in Eq. equation 2, PPO is widely used in practice.

4 Lifelong RLHF

In the previous section (Sec. 3), we outlined the RLHF procedure to align LLMs using a preference dataset. The procedure assumes that the prompt distribution and preferences generated by humans (or AI) are static; however, in practice, the prompt distribution and preferences of individuals change over time. For instance, in the LLM tutoring application, the difficulty of the questions (or the nature of hints) generated by the LLM tutor varies as the student learns the subject. In such cases, the LLM agent must continually adapt to reflect the latest preferences of an individual: *lifelong RLHF*.

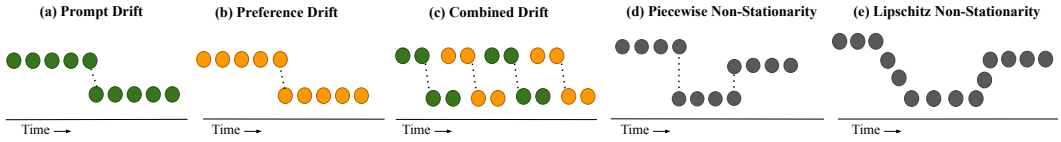


Figure 2: Non-stationarities in RLHF: (a-c) modes of drift and (d-e) types of drift. *(a)* Only the prompt distribution changes. *(b)* Only the preference distribution changes. *(c)* Both the prompt and the preference distributions change. *(d)* Piecewise non-stationarity. *(e)* Lipschitz non-stationarity.

In lifelong RLHF, the goal at each time step, $t = 0, 1, 2, \ldots$, is to align the LLM using a new preference dataset batch, $D_t = \{x^i, y_c^i, y_r^i\}_{i=1}^{N_t}$, while retaining useful prior knowledge to accelerate future adaptation. This introduces the possibility of incorporating a KL term into the Lifelong RLHF objective, $\beta \mathbb{D}_{KL}(\pi_{\theta_t}(\cdot|x)||\pi_{\theta_{t-1}}(\cdot|x))$, akin to (Zhang et al., 2024b), with the key distinction that this divergence need not be enforced when human preferences undergo significant shifts. At each step, prompts are drawn from a time-dependent distribution p_t, and preferences are generated via a reward function $r_t^*(y, x)$. N_t is the number of preference samples at time t.

From one time step to another, the prompt distribution, the underlying reward function, or both can change, *modes of drift*:

- **Prompt drift:** The prompt distribution changes, $p_t \neq p_{t'}$;
- **Preference drift:** The underlying reward function from which the preferences are generated changes, $r_t^*(y, x) \neq r_{t'}^*(y, x)$;
- **Combined drift:** Both the prompt distribution and the underlying reward function change.

Following the literature on lifelong RL (Khetarpal et al., 2022; Abel et al., 2023), we consider two ways in which various modes of drift can evolve over time, *types of drift*:

- **Piecewise non-stationarity:** When the change is sudden in one of the three modes of drift. For example, piecewise preference drift is:

$$r_t^*(y, x) = \begin{cases} r^0(y, x), & 0 \leq t \leq t_0, \\ r^1(y, x), & t_0 < t \leq t_1, \\ \vdots \end{cases}$$

- **Lipschitz non-stationarity:** When the change is gradual in any of the three modes of drift. The Lipschitz preference drift is:

$$\left|r_t^*(y, x) - r_{t'}^*(y, x)\right| \leq C\,|t - t'|, \quad \forall x, y, t, t',$$

  where $C > 0$ is the Lipschitz constant that determines the rate of change

Figure 2 shows the three modes and the two types of drift discussed here. Although there are several other types of drift, we restrict our study to piecewise and Lipschitz non-stationarities due to their simplicity and broad applicability. We next show how previously introduced lifelong RLHF problems can be viewed as special cases of our framework.

Remark 1. The lifelong RLHF problem introduced by Zhang et al. (2024b) for CPPO is prompt drift under piecewise non-stationary.

Proof. In CPPO, a task from a sequence has two datasets: a human feedback dataset containing information about the chosen and the rejected responses, and a prompt dataset. Since their objective function, $\max_{\pi_\theta} \sum_{t=1}^{T} \mathbb{E}_{x\sim p_t(x), y\sim\pi_\theta(\cdot|x)}[r_t(y, x)]$, maximizes all rewards from the past, preferences are implicitly static (TL;DR summarization). So, only the prompt distribution changes from one task to another, implying *prompt drift*. Since the datasets for the two consecutive tasks are disjunct (different subreddits) and can be arbitrarily different, we can classify it as *piecewise non-stationarity*. □

5 AIF-GEN PLATFORM

Collecting human annotations in a continual setting is resource-intensive and time-consuming. To address this, inspired by the recent success of synthetic data for single-task alignment, we introduce

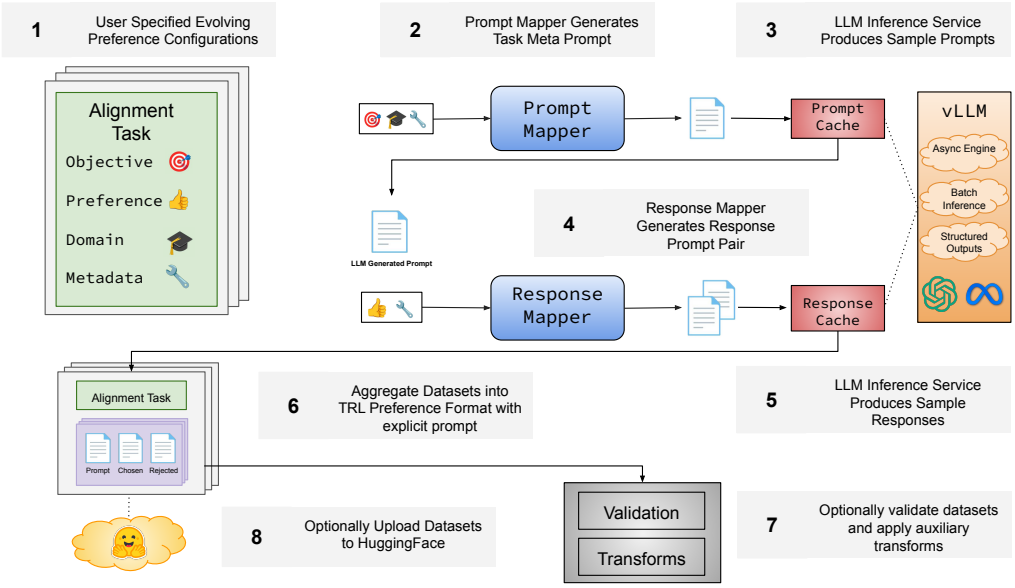


Figure 3: High-level AIF-GEN design. (1) Users specify RLHF tasks (objectives, preferences, domains) and metadata in YAML configs. (2) Prompt Mapper generates meta-prompts, which (3) produce sample prompts using an LLM. (4) Response Mapper pairs these with preferences to create response prompts, which (5) generate *chosen* and *rejected* outputs. Samples are aggregated (6), and optionally validated, transformed (7), or (8) uploaded to HuggingFace.

AIF-GEN—a software platform designed to generate synthetic preference data for various *modes of drift* and *types of non-stationarity* discussed in Section 4. AIF-GEN (see Figure 3) allows users to define a sequence of *Alignment Tasks* (§5.1) in structured YAML files that encode evolving *objectives, domains, and preferences*. These configurations serve as inputs to an asynchronous inference engine powered by vLLM Kwon et al. (2023), which generates prompt–response pairs at scale. Prompt templates are managed by the *Prompt Mapper*, while the *Response Mapper* synthesizes candidate completions and formats the final preference samples (§5.2). To support downstream workflows, the platform also includes a CLI tool for dataset validation (e.g., computing sample diversity with embedding models), transformation (e.g., merging multiple datasets), and publishing (e.g., uploading to Hugging Face).

5.1 Simulating Non-Stationarity

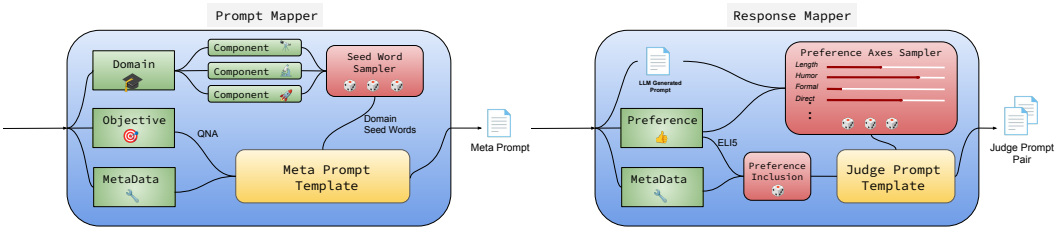
To enable a systematic study of non-stationarity in RLAIF, each alignment task in AIF-GEN is decomposed into 3 orthogonal components: *Objective*, *Domain*, and *Preference*. This decomposition allows researchers to isolate and manipulate distinct modes of drift. Specifically:

Objective: the underlying task, such as question answering or summarization.

Domain: the distribution over prompts as a controllable mechanism for inducing domain shifts. Users specify domains with seed word vocabularies—tokens relevant to particular topics/subfields (e.g., biology or world history). These are sampled and injected into templates to simulate prompt drift.

Preference: the latent reward signal—i.e., the desirable output given a task and domain. These include styles like "explain like I'm five" and are critical for simulating shifts in user intent. Preferences guide the reward modelling process and influence which responses are ranked as preferred.

Users can simulate diverse non-stationarities by varying these components across a sequence, such as domain shifts under a fixed task or evolving stylistic preferences. For example, one might hold the objective (Q&A) constant while progressing from arithmetic to calculus domains, with preferences shifting from "concise" to "detailed explanations". We publicly release all configuration files used in our experiments, with representative examples in the appendix. These provide a flexible foundation for defining custom alignment scenarios, supporting the systematic exploration of lifelong RLHF.



(a) *Prompt Mapper* internals. (b) *Response Mapper* internals.

Figure 4: (a) The user-defined domain is decomposed into semantic components, each mapped to seed word vocabularies. These are sampled and composed with the task objective using a meta-prompt template. Metadata allows custom prefixes/suffixes in the prompt. (b) The task preference and LLM-generated prompt are passed to the judge template. Auxiliary styles are sampled via preference axes to diversify outputs. The task preference is optionally included to calibrate the difficulty of distinguishing preferred responses. Metadata can modulate preference axes, allowing users to adjust the reward modelling task's difficulty by controlling each response's subtlety.

5.2 Mapper Internals

Prompt Mapper internals are shown in Figure 4a. For each sample, domain-specific seed words are randomly drawn to promote prompt diversity. These vocabularies—published with our code—are fully configurable, allowing users to define domains like education with subfields (e.g., astronomy, biology, engineering). Seed words are combined with the task objective (e.g., Q&A) to generate a Meta Prompt, which is passed to the LLM. Templates are provided in the Appendix.

Response Mapper internals are shown in Figure 4b. Each prompt (e.g., a biology question) is paired with auxiliary styles (e.g., length, humour) to generate diverse responses. The task preference (e.g., "explain like I'm five") is probabilistically included (configurable) to influence the Judge Prompt. This helps calibrate the difficulty of distinguishing preferred responses. Resulting prompts are sent to the inference engine to produce chosen/rejected pairs. Templates are provided in the Appendix.

6 Datasets

6.1 Data Generation

We used `AIF-GEN` with GPT-4o-mini to generate our data. We experimented with generation using Llama 3.2 70b (Grattafiori et al., 2024) for validation purposes only, and we don't release this dataset. All data was generated with a temperature of 0.99, a maximum prompt length of 1024 tokens, and a response cap of 2048 tokens.

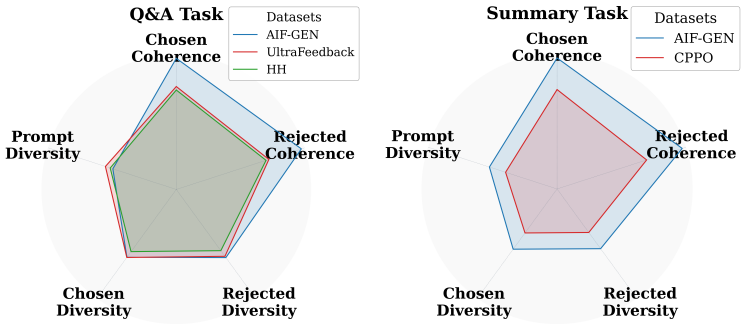
Static Datasets. We first generated static datasets—valuable on their own for evaluating alignment methods. We define each alignment task by specifying an objective, domain, and preference. For objectives, we include *question answering*, *summarization*, and *text generation*. Domains include *education*, *politics*, *tech/healthcare*, and *tech/physics*, using domain-specific seed word vocabularies (released with our code). We sampled two seed words per prompt. The response mapper was configured to sample 3 styles per response with a preference inclusion probability of 0.4. Preferences were stylistic instruction such as *ELI5*, *expert*, *formal*, and *Shakespearean*. Details of the preferences and templates are included in the appendix. Each static dataset has roughly 10,000 samples.

Continual Datasets. To evaluate alignment under drift, we constructed 4 continual datasets by merging subsets of the static data. Our platform also allows for generating non-stationary datasets natively. Each dataset simulates a different type of non-stationarity through structured transitions across preference, domain, and objective. The *Lipschitz* dataset progressively shifts preference complexity by concatenating tech/physics summarization samples across three levels: *ELI5* $\rightarrow$ *high school* $\rightarrow$ *expert*. This simulates a gradual increase in difficulty while keeping the domain and objective fixed. The *Piecewise Preference* dataset cycles through stylistic preference shifts—*rapper*, *Shakespearean*, and *formal*—within political generation tasks, repeated over 3 cycles to model recurring non-stationarity. To explore the effect of non-stationary frequency, we created a Piecewise Q&A dataset alternating

between *hinted* and *directed* Q&A with 5000 examples from each static subset. Finally, the *Mixed* dataset combines simultaneous shifts across objective, domain, and preference. It transitions through education Q&A (*ELI5*), education Q&A (*expert*), political summarization (*ELI5*), and tech/healthcare Q&A (*expert*). This dataset is the most challenging with multi-axis drift, ideal for stress-testing lifelong RLHF algorithms.

6.2 LLM Validation

Figure 5 compares the quality of datasets generated by AIF-GEN against prior work, using two key metrics: Coherence and Diversity. Coherence captures the logical consistency of responses and is scored (0–10) by GPT-4o-mini acting as an LLM judge. Diversity measures the variation across responses, computed using the average pairwise cosine distance of embeddings from the Salesforce/SFR-Embedding-Mistral model (Meng et al., 2024) (details in Appendix). For a fair comparison, we group datasets into Q&A and summarization tasks. As shown, AIF-GEN consistently produces higher-coherence outputs and more diverse prompts and responses. Because AIF-GEN is LLM-agnostic, coherence and diversity are expected to improve as stronger base models are used for generation.



(a) Q&A radar plot.

(b) Summary radar plot.

Figure 5: AIF-GEN generates higher quality RLHF datasets for both Q&A and summarization tasks compared to previous datasets.

6.3 Human Evaluation

To assess the quality of our data, we conducted a targeted human evaluation on a subset of the education Q&A domain using 3 stylistic preferences: Hinted, ELI5 and Expert. These were chosen to span a spectrum of alignment difficulty. We randomly sampled 50 prompt–response pairs per preference and got 3 independent annotators to select the response better aligned with the stated preference. Annotators could also choose *both* or *neither* if responses were equally aligned or unaligned, and flag incoherent samples (e.g., ill-formed prompts). While limited in scope, this evaluation confirms AIF-GEN's ability to generate preference-aligned data across diverse styles.

Table 2 reports four metrics across preference types: (1) Unanimous Consensus Rate—full annotator agreement, (2) Fleiss' Kappa—a standard measure of inter-rater reliability, (3) LLM Judge Agreement—match between LLM and human majority vote, and (4) Inter-Human Agreement—average annotator alignment with the majority. *Both* and *neither* human labels were randomly mapped to binary labels for consistency with LLM outputs. Metrics include 95% confidence intervals from 1,000 bootstrap samples. Only 5 of 450 samples were flagged as incoherent, due to rare seed word combinations. Results show 48–64% unanimous consensus and strong Kappa scores (≈0.5 for ELI5/Expert, 0.31 for Hinted), which fall within accepted bounds for solid inter-rater agreement, with the lower score on Hinted reflecting its inherently more subtle nature. LLM Judge Agreement hovers near 60% across all three preferences, suggesting non-trivial but comparable difficulty to results reported in existing datasets (Cui et al., 2023). Figure 6 shows a 2×4 confusion matrix comparing LLM judgments with human labels across all preference types. Most LLM outputs match the human

Metric	Hinted	ELI5	Expert
Unanimous Consensus Rate	0.48	0.64	0.62
Fleiss' Kappa	0.31	0.52	0.49
LLM Judge Agreement	0.64	0.56	0.58
Inter-Human Agreement	0.83	0.88	0.87

Table 2: Education Q&A Human Evaluation

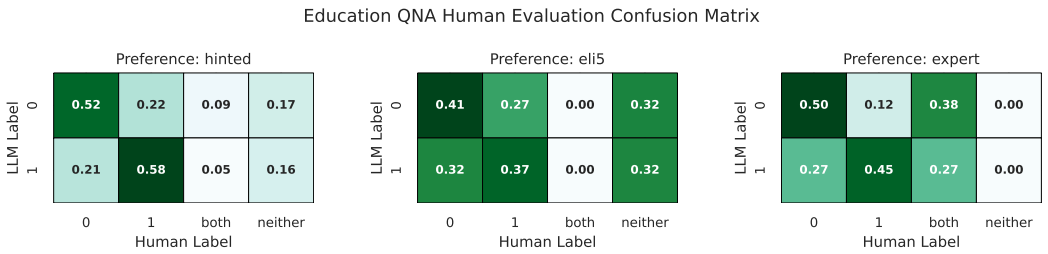


Figure 6: Confusion matrix comparing LLM (response 0 or 1) with human annotations (response 0, 1, *both*, *neither*). Most samples show agreement. When humans label *both* or *neither*, the LLM's predictions become more evenly split, reflecting uncertainty.

majority's choices. When annotators select *both* or *neither*, LLM responses become more evenly split, reflecting lower confidence. Notably, ELI5 shows more *neither* votes, while Expert has more *both*, suggesting ELI5 may be harder to model and could benefit from stronger preference conditioning in generation.

7 EXPERIMENTS

Our goal here is to demonstrate how the AIF-GEN platform and its accompanying datasets (Sections 5 and 6.1) can be effectively used to study lifelong RLHF. To accomplish this, we use 4 continual datasets discussed in Section 6.1 for the experiments. We consider three algorithms with different properties: (a) DPO (Rafailov et al., 2024), which optimizes the policy using the preference data directly without constructing an intermediate reward model; (b) PPO (Schulman et al., 2017) which is a widely used RL algorithm that optimizes a learned reward model; (c) CPPO (Zhang et al., 2024b) which is the only published continual RLHF algorithm. We use the Qwen2 0.5B model (Yang et al., 2024) as our base model for alignment and reward training. Here we present the results and analysis for DPO, while additional experiments with PPO and CPPO are included in the Appendix. Comprehensive experimental details, including hyperparameters and hardware specifications, are also provided in the Appendix.

Since DPO models are trained directly on preference pairs, we evaluate their performance using train and test accuracies. Accuracy is measured as the fraction of examples where the chosen response is preferred over the rejected one, according to the implicit DPO reward model. To provide a more comprehensive evaluation, we include additional metrics—reward model scores and forgetting measures—in the Appendix.

Observations: Across all tasks, we find that DPO performs well on the training distribution, but test accuracy consistently lags—indicating overfitting to observed preference pairs. Interestingly, DPO retains some memory of prior tasks in the training set, particularly in structured, recurring settings. For instance, in the domain preference shift task, where the prompt transitions from ELI5 to expert and then cycles back, we observe a smaller performance drop during the second transition. This trend appears on train and test sets, suggesting that DPO may benefit from prior exposure when domain shifts repeat. A similar pattern emerges in the piecewise preference shift benchmark, where sub-tasks (e.g., rapper, Shakespeare, formal) reappear multiple times. After the first complete cycle, DPO maintains stable training performance across switches, likely due to memorization. However, the test accuracy dips each time a sub-task changes, indicating that while the model can retain patterns it has seen, it struggles to generalize to held-out examples from previous preferences.

These observations paint a nuanced picture: DPO exhibits partial continual learning behaviour, particularly on training data, but fails to generalize across dynamic preference distributions. This limitation underscores a key gap in current RLHF algorithms—they lack mechanisms to retain and adapt preferences in a principled way under non-stationarity. Our results motivate the development of continual RLHF algorithms that generalize across evolving tasks, not just memorize them. In this context, our proposed AIF-GEN platform and suite of lifelong learning datasets provide a valuable testbed for diagnosing failure modes and accelerating algorithmic progress.

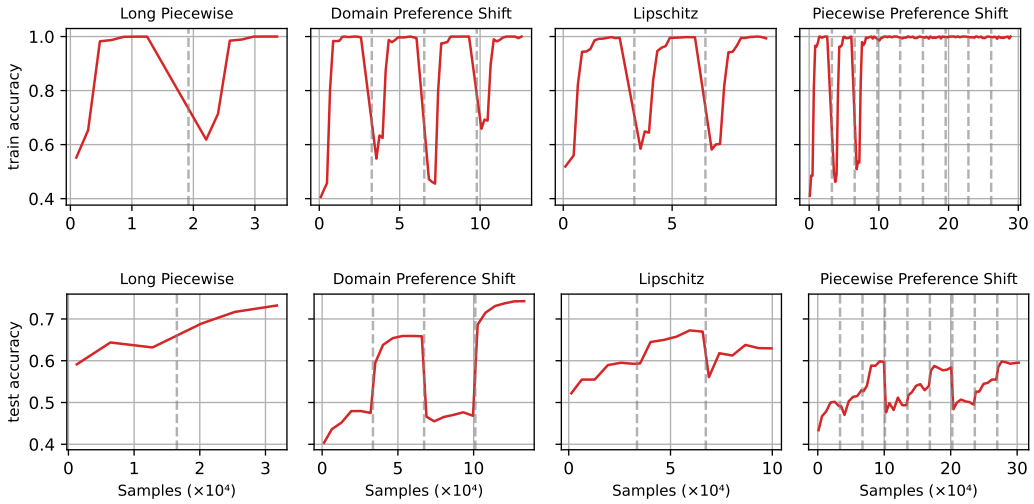


Figure 7: DPO train and test accuracy during training. Dashed lines denote task switches in the continual learning datasets.

8 Discussion and Limitations

The success of reinforcement learning has been driven in large part by open-source simulators like Atari (Bellemare et al., 2013) and MuJoCo (Todorov et al., 2012), which enabled rapid development and benchmarking of core algorithms such as DQN (Mnih et al., 2013) and PPO (Schulman et al., 2017). In contrast, progress in lifelong RLHF remains bottlenecked by the lack of scalable, time-evolving human preference data. To address this gap, we introduce `AIF-GEN`, a synthetic preference generation platform that plays a similar role to simulators in RL—supporting the development and evaluation of continual alignment algorithms across diverse tasks and evolving user preferences.

Through experiments, we analyzed DPO, PPO, and CPPO algorithms under various kinds of non-stationary problems, synthesized by `AIF-GEN`. In future, building on top of these insights, we expect the community to develop novel lifelong RLHF algorithms. Our tool can also serve as a standardized benchmark for advancing reproducible research in single-task and lifelong RLHF. While our software platform is catered for generating tasks for aligning LLMs, similar tools can be developed to incorporate other modalities like vision, to advance lifelong alignment research in vision-language models (Li et al., 2019; Lu et al., 2019).

While `AIF-GEN` provides a flexible foundation for lifelong RLHF, several design choices warrant discussion. First, our prompt templates, though diverse, are inherently handcrafted. However, this reflects a core strength: users can easily define or extend their own structured configurations to suit new domains, preference styles, and task formats. Second, we do not study model scaling effects. We focus on GPT-4o-mini for data generation due to its strong instruction-following capabilities, competitive quality relative to larger models, and lower computational cost, making it a practical and reproducible choice for large-scale synthetic data creation. Since `AIF-GEN` is model-agnostic, future work can readily integrate emerging open or closed models. Third, while our automated evaluations use LLM judges with potential biases, targeted human studies confirm overall data quality. Due to resource constraints, we did not evaluate all datasets. However, `AIF-GEN` is a platform, not a fixed benchmark—datasets and inturn the performance of algorithms on these datasets will depend on user-defined templates, models, and objectives. We see this adaptability as a key feature for advancing lifelong alignment research.

9 Conclusion

Aligning LLMs with evolving human preferences is a central challenge for real-world deployment. We introduce lifelong RLAIF and present `AIF-GEN`, the first platform for generating dynamic, preference-driven datasets for continual RLHF. Built on vLLM with structured prompts and LLM-

agnostic APIs, AIF-GEN enables scalable simulation of alignment drift across tasks, domains, and styles. Our datasets—validated through human evaluations—allow a systematic study of non-stationarity in reward learning. Experiments with PPO, DPO, and CPPO reveal that adaptation strategies must be context-sensitive. We envision AIF-GEN as a foundation for the next generation of alignment benchmarks, accelerating progress toward truly adaptive, lifelong language models.

REFERENCES

David Abel, André Barreto, Benjamin Van Roy, Doina Precup, Hado van Hasselt, and Satinder Singh. A definition of continual reinforcement learning, 2023. URL https://arxiv.org/abs/2307.11046.

Reza Yazdani Aminabadi, Samyam Rajbhandari, Minjia Zhang, Ammar Ahmad Awan, Cheng Li, Du Li, Elton Zheng, Jeff Rasley, Shaden Smith, Olatunji Ruwase, and Yuxiong He. Deepspeed inference: Enabling efficient inference of transformer models at unprecedented scale, 2022. URL https://arxiv.org/abs/2207.00032.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv: 2212.08073*, 2022.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of artificial intelligence research*, 47: 253–279, 2013.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Paul F. Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 4299–4307, 2017.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: Boosting language models with high-quality feedback, 2023.

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe rlhf: Safe reinforcement learning from human feedback, 2023. URL https://arxiv.org/abs/2310.12773.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie

Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie

Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Irina Jurenka, Markus Kunesch, Kevin R. McKee, Daniel Gillick, Shaojian Zhu, Sara Wiltberger, Shubham Milind Phal, Katherine Hermann, Daniel Kasenberg, Avishkar Bhoopchand, Ankit Anand, Miruna Pîslar, Stephanie Chan, Lisa Wang, Jennifer She, Parsa Mahmoudieh, Aliya Rysbek, Wei-Jen Ko, Andrea Huber, Brett Wiltshire, Gal Elidan, Roni Rabin, Jasmin Rubinovitz, Amit Pitaru, Mac McAllister, Julia Wilkowski, David Choi, Roee Engelberg, Lidan Hackmon, Adva Levin, Rachel Griffin, Michael Sears, Filip Bar, Mia Mesar, Mana Jabbour, Arslan Chaudhry, James Cohan, Sridhar Thiagarajan, Nir Levine, Ben Brown, Dilan Gorur, Svetlana Grant, Rachel Hashimshoni, Laura Weidinger, Jieru Hu, Dawn Chen, Kuba Dolecki, Canfer Akbulut, Maxwell Bileschi, Laura Culp, Wen-Xin Dong, Nahema Marchal, Kelsie Van Deman, Hema Bajaj Misra, Michael Duah, Moran Ambar, Avi Caciularu, Sandra Lefdal, Chris Summerfield, James An, Pierre-Alexandre Kamienny, Abhinit Mohdi, Theofilos Strinopoulous, Annie Hale, Wayne Anderson, Luis C. Cobo, Niv Efron, Muktha Ananda, Shakir Mohamed, Maureen Heymans, Zoubin Ghahramani, Yossi Matias, Ben Gomes, and Lila Ibrahim. Towards responsible development of generative ai for education: An evaluation-driven approach, 2024. URL https://arxiv.org/abs/2407.12687.

Jared Kaplan et al. Training a helpful and harmless assistant with reinforcement learning from human feedback, April 2022. URL https://arxiv.org/abs/2204.05862.

Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. Towards continual reinforcement learning: A review and perspectives. *Journal of Artificial Intelligence Research*, 75:1401–1476, 2022.

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

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019.

Xinyu Li and Yu Chen. Reinforcement learning from ai feedback. *arXiv preprint arXiv:2305.12345*, 2023.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019.

Ryan Marten, Trung Vu, Charlie Cheng-Jie Ji, Kartik Sharma, Shreyas Pimpalgaonkar, Alex Dimakis, and Maheswaran Sathiamoorthy. Curator: A tool for synthetic data creation. https://github.com/bespokelabsai/curator, January 2025.

Ian R. McKenzie, Alexander Lyzhov, Michael Pieler, Alicia Parrish, Aaron Mueller, Ameya Prabhu, Euan McLean, Aaron Kirtland, Alexis Ross, Alisa Liu, Andrew Gritsevskiy, Daniel Wurgaft, Derik Kauffman, Gabriel Recchia, Jiacheng Liu, Joe Cavanagh, Max Weiss, Sicong Huang, The Floating Droid, Tom Tseng, Tomasz Korbak, Xudong Shen, Yuhui Zhang, Zhengping Zhou, Najoung Kim, Samuel R. Bowman, and Ethan Perez. Inverse scaling: When bigger isn't better, 2024. URL https://arxiv.org/abs/2306.09479.

Rui Meng, Ye Liu, Shafiq Rayhan Joty, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. Sfr-embedding-mistral:enhance text retrieval with transfer learning. Salesforce AI Research Blog, 2024. URL https://www.salesforce.com/blog/sfr-embedding/.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Rémi Munos, Michal Valko, Daniele Calandriello, Mohammad Gheshlaghi Azar, Mark Rowland, Zhaohan Daniel Guo, Yunhao Tang, Matthieu Geist, Thomas Mesnard, Andrea Michi, Marco Selvi, Sertan Girgin, Nikola Momchev, Olivier Bachem, Daniel J. Mankowitz, Doina Precup, and Bilal Piot. Nash learning from human feedback, 2024. URL https://arxiv.org/abs/2312.00886.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, and et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 27730–27744, 2022.

German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

Ajay Patel, Colin Raffel, and Chris Callison-Burch. DataDreamer: A tool for synthetic data generation and reproducible LLM workflows. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3781–3799, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.208. URL https://aclanthology.org/2024.acl-long.208/.

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

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024. URL https://arxiv.org/abs/2305.18290.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. In *Advances in Neural Information Processing Systems*, volume 30, pp. 1133–1143, 2017.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, and et al. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 3008–3021, 2020.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback, 2022. URL https://arxiv.org/abs/2009.01325.

Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liang-Yan Gui, Yu-Xiong Wang, Yiming Yang, Kurt Keutzer, and Trevor Darrell. Aligning large multimodal models with factually augmented rlhf, 2023. URL https://arxiv.org/abs/2309.14525.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. Trl: Transformer reinforcement learning. https://github.com/huggingface/trl, 2020.

Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. Iterative preference learning from human feedback: Bridging theory and practice for rlhf under kl-constraint, 2024. URL https://arxiv.org/abs/2312.11456.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024. URL https://arxiv.org/abs/2407.10671.

Han Zhang, Lin Gui, Yu Lei, Yuanzhao Zhai, and et al. Copr: Continual human preference learning via optimal policy regularization. *arXiv preprint arXiv:2402.14228*, 2024a.

Han Zhang, Yu Lei, Lin Gui, Min Yang, Yulan He, Hui Wang, and Ruifeng Xu. CPPO: Continual learning for reinforcement learning with human feedback. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=86zAUE80pP.

Qiang Zhang, Xia Li, and Yang Chen. Synthetic data generation for reinforcement learning: A survey. *arXiv preprint arXiv:2301.01234*, 2023.

Banghua Zhu, Jiantao Jiao, and Michael I. Jordan. Principled reinforcement learning with human feedback from pairwise or k-wise comparisons, 2024. URL https://arxiv.org/abs/2301.11270.

APPENDIX

A LIFELONG RLHF

A.1 PSEUDOCODE

Algorithm 1 Lifelong RLHF

Initialize: $\pi_\theta \leftarrow \pi^{SFT}$
for each time step $t = 1, 2, \ldots$ **do**
    Collect batch of data $\mathcal{D}_t$ generated by Algorithm 1
    Fit a reward model $r_{\phi,t}$ on $\mathcal{D}_t$
    Update policy (LLM) π_θ using your favourite algorithm e.g. PPO
end for

Algorithm 1 outlines the high-level pseudocode for the Lifelong RLHF training loop. We begin by initializing the LLM policy weights with a supervised fine-tuned (SFT) model, $\pi_\theta \leftarrow \pi^{SFT}$. At each time step t, we iteratively collect preference data and update the policy. Specifically, we generate a dataset $\mathcal{D}_t$ using the procedure described in Algorithm 2, which captures preferences over model outputs in the current task context. A reward model $r_{\phi,t}$ is then trained on $\mathcal{D}_t$ to model these preferences. Using $r_{\phi,t}$, we update the parameters θ of an LLM by optimizing the objective of your favourite algorithm as detailed in Section A.2.

Algorithm 2 Synthetic Preference Generation for Task T_t

Input: LLM $\mathcal{M}$, Budget N_t, LLM Templates for prompt generation τ_{prompt}, response generation τ_{response}, and preference ranking τ_{judge}
Initialize $\mathcal{D}_t \leftarrow \emptyset$
for $i = 1$ to N_t **do**
    Generate a prompt $x \sim \mathcal{M}(\cdot|\tau_{\text{prompt}})$
    Generate two responses
$$y_1,\, y_2 \;\sim\; \mathcal{M}(\,\cdot \mid x, \tau_{\text{response}})$$
    Determine preference labels
$$y_c,\, y_r \;\sim\; \mathcal{M}(\,\cdot \mid x, y_1, y_2, \tau_{\text{judge}})$$
    Append $\langle x, y_{\text{c}}, y_{\text{r}} \rangle$ to $\mathcal{D}_t$
end for
return $\mathcal{D}_t$

Algorithm 2 outlines the procedure for generating synthetic preference data at each temporal phase of continual learning. We assume access to an LLM $\mathcal{M}$, which generates prompts and corresponding responses. While separate models could be employed for prompt and response generation, we assume a single model for simplicity. At each time step t, the objective is to construct a dataset $\mathcal{D}_t$ of synthetic preference samples, given a (latent) prompt distribution $p_t(x)$ and a compute budget of N_t queries. We also assume access to LLM templates: τ_{prompt} for prompt generation, τ_{response} for response generation, and τ_{judge} for preference selection—each specified via configuration files.

We initialize $\mathcal{D}_t$ as empty and iterate N_t times. Since the true distribution $p_t(x)$ is inaccessible, we approximate it using the prompt templates τ_{prompt}, which encode domain and task-specific characteristics. In each iteration, the LLM $\mathcal{M}$ is conditioned on τ_{prompt} to generate a sample prompt x, then queried again to sample two responses $y_1, y_2 \sim \mathcal{M}(\cdot \mid x, \tau_{\text{response}})$. A final inference step applies the judge templates τ_{judge} to select the preferred and rejected responses, denoted y_{c} and y_{r}. The resulting preference-labeled tuple $\langle x, y_{\text{c}}, y_{\text{r}} \rangle$ is added to $\mathcal{D}_t$. Although this procedure produces binary preference data, it can be extended to more expressive preference formats, such as listwise comparisons or multi-response rankings.

A.2 ALGORITHMS

Recall our general continual RLHF objective at time t:

$$\max_{\pi_\theta} \mathbb{E}_{x\sim D, y\sim\pi_\theta(\cdot|x)}[r_\phi(y,x) - \beta\mathbb{D}_{KL}(\pi_\theta(\cdot|x)||\pi^{SFT}(\cdot|x))] \tag{3}$$

Here π^{SFT} is a fixed reference policy (e.g. the SFT model), and r_ϕ is the reward model trained on D.

1. PROXIMAL POLICY OPTIMIZATION (PPO)

PPO optimizes policies by approximating the KL-divergence constraint with a clipped surrogate objective, promoting stability by preventing excessively large updates. It balances exploration and exploitation, maintaining efficiency by controlling the update size through a hyperparameter ϵ.

PPO replaces the exact KL-penalty in Eq. equation 3 with a *clipped* surrogate:

$$r_t(\theta) = \frac{\pi_\theta(y \mid x)}{\pi^{SFT}(y \mid x)}, \quad A_t = r_{\phi,t}(x,y). \tag{4}$$

The PPO surrogate objective at time t is

$$\mathcal{L}_t^{PPO} = \mathbb{E}_{x,y}\Big[\min\big(r_t(\theta)\,A_t,\ \text{clip}\big(r_t(\theta),\,1-\epsilon,\,1+\epsilon\big)\,A_t\big)\Big]. \tag{5}$$

When ϵ is small, this approximates a trust-region/KL-constrained solve of Eq. equation 3 with $\beta \approx 1/\epsilon$.

A_t is the advantage function, defined as $A_t = r_{\phi,t}(x,y)$, which measures how much better an action y is compared to the expected reward for input x.

2. DIRECT PREFERENCE OPTIMIZATION (DPO)

DPO directly optimizes policy performance using pairwise human preference comparisons, eliminating explicit KL penalties. It leverages a logistic function for pairwise classification, optimizing policy parameters toward responses preferred by humans. This simplification avoids explicit reward modelling complexity and KL constraints by treating preferences as binary feedback signals, but the updates are specific to preferences from the Bradley-Terry model. DPO (Rafailov et al., 2024) optimizes directly over pairwise comparisons $\{(x, y_c, y_r)\}$ by

$$\mathcal{L}_t = -\mathbb{E}_{(x,y_c,y_r)\in D_t}\left[\log\sigma\Big(\beta\log\frac{\pi_\theta(y_c \mid x)}{\log\pi^{SFT}(y_c \mid x)} - \beta\log\frac{\pi_\theta(y_r \mid x)}{\log\pi^{SFT}(y_r \mid x)}\Big)\right]. \tag{6}$$

where σ is the logistic function.

In the above equation, π_θ denotes the current policy parameterized by θ, and π^{SFT} is a fixed reference policy. The variable x represents the input prompt, while y_c and y_r denote the preferred (chosen) and less preferred (reject) responses. These comparisons are drawn from a dataset $\mathcal{D}$ consisting of triplets (x, y_w, y_l). The scalar β is a temperature-like hyperparameter that controls the deviation from the base reference policy (SFT model). The objective leverages the logistic sigmoid function $\sigma(\cdot)$ to compute the probability that the model prefers y_c over y_r, based on the log-probability ratio of each response under π_θ relative to π^{SFT}.

3. CONTINUAL PPO (CPPO)

CPPO iteratively applies PPO optimization to new data streams, continuously adapting the policy while maintaining previous learned knowledge by constraining the current policy to be close to the previously learned policy using an additional KL divergence term. While the PPO objective learns from the new datastream, the additional KL term preserves older knowledge, thereby efficiently trading off stability (preserving past knowledge) and plasticity (adapting to new data) in continual alignment tasks. At each time step, CPPO's objective is:

$$\mathcal{L}_t = \mathcal{L}_t^{PPO} - \mathbb{E}_{(x,y_c,y_r)\in\hat{D}}\left[\,\mathbb{D}_{KL}\left(\pi_\theta(\cdot \mid x)\,\|\,\pi_{t-1}(\cdot \mid x)\right)\right], \tag{7}$$

where $\mathcal{L}_{\mathcal{PPO}}$ is the PPO loss that is defined in Eq. 5, $\hat{D}$ is a specially curated dataset using the samples from all tasks until the current time step t, and $\pi_{t-1}(\cdot \mid x)$ is the policy at the end of the previous task.

B Experiment Details and Compute Resources

We implemented and benchmarked three state-of-the-art RLHF and Continual RLHF algorithms: PPO, DPO, and CPPO (Schulman et al., 2017; Zhang et al., 2024b; Rafailov et al., 2024). Our implementation adapts TRL's (Transformers Reinforcement Learning (von Werra et al., 2020)) framework for PPO and DPO, while our CPPO implementation extends the original authors' codebase. All implementations were optimized for distributed training across multiple GPUs using DeepSpeed ZeRO-2/3 (Aminabadi et al., 2022), enabling efficient fine-tuning even with limited computational resources. The continual learning approach allows sequential adaptation to domain shifts while maintaining performance on previously seen tasks. Based on dataset lengths, experiments compute usage and time efficiency vary; however, an average experiment is done on four NVIDIA H100 Tensor Core GPUs for 12 hours, resulting in 48 H100 GPU hours. Tables 3, 4, 5, and 6 respectively display the set of hyperparameters used to fine-tune the Qwen 2 0.5B model (Yang et al., 2024).

Table 3: Reward Model Hyperparameters

Hyperparameter	**Value**
Learning rate	1.0×10^{-5}
Training epochs	3
Batch size (per device)	8
Loss function	Binary cross-entropy
Label smoothing	0.1
Weight decay	0.01
Warmup ratio	0.1
Mixed precision	BF16
Gradient checkpointing	Enabled

Table 4: PPO Hyperparameters

Hyperparameter	**Value**
Learning rate	1.0×10^{-6}
KL coefficient	0.37
PPO clip range	0.1
Response length	256
Training epochs	4
Batch size (per device)	16
Mixed precision	BF16
Gradient checkpointing	Enabled

Table 5: DPO Hyperparameters

Hyperparameter	**Value**
Learning rate	5.0×10^{-6}
Training epochs	4
Batch size (per device)	8
Response length	256
Mixed precision	BF16
Gradient checkpointing	Enabled

Table 6: CPPO Hyperparameters

Hyperparameter	**Value**
Learning rate	1.0×10^{-6}
KL coefficient	0.37
PPO clip range	0.1
Response length	256
Training epochs	4
Batch size (per device)	16
Mixed precision	BF16
Gradient checkpointing	Enabled

C Additional Experiments

We train separate reward models for each task in the continual datasets using a frozen version of the base model as the scorer. The appropriate reward model is dynamically selected during PPO training based on the current task.

C.1 Reward Models

We designed `AIF-GEN` to blur the line between chosen and rejected responses, making reward modelling especially challenging. In contrast to UltraFeedback and HH-RLHF—where reward models attain high test accuracy—models fine-tuned on Qwen 2 0.5B achieve only 70% accuracy on `AIF-GEN`, underscoring its difficulty. Though CPPO matches this classification challenge, `AIF-GEN` covers a broader array of preferences, domains, and objectives, which amplifies ambiguity

and thwarts models that rely on superficial reward cues. Our findings imply that existing reward-modelling approaches—whether static or lifelong—must be retooled for datasets with minimal divergence between positive and negative examples. We defer investigation of model scaling effects to future work; prior evidence suggests that larger models do not always yield proportional gains in reward modelling (McKenzie et al., 2024).

Reward-based evaluations are performed using held-out reward models corresponding to each task segment.

C.2 PPO AND CPPO

We distinguish DPO from PPO-style algorithms (PPO and CPPO) because their training objectives and dependencies differ fundamentally. DPO does not rely on a learned reward model; its optimization is driven directly by human preference data, and its evaluation is typically framed in terms of accuracy against reference labels. In contrast, PPO and CPPO both incorporate a reward model into their policy updates and are therefore subject to its biases. As a result, we present DPO results in a separate section—where the focus is on accuracy metrics—while PPO and CPPO are compared in a reward-score context. This separation prevents conflating differences that arise purely from the presence or absence of a learned reward function.

Figure 8 summarizes our comparison of PPO versus CPPO across five continual-learning datasets, AIF-GEN's and CPPO's datasets. Both algorithms exhibit high variance after mastering the first task on the long-piecewise benchmark, but PPO maintains its peak performance rather than regressing. In the domain-preference-shift scenario, CPPO achieves higher average training scores, suggesting improved adaptability—yet in the held-out evaluation, its performance converges to that of PPO, indicating similar generalizations. The Lipschitz dataset yields nearly identical variance profiles and accuracy/reward trajectories for both methods throughout training and testing, underscoring that CPPO's design for knowledge retention does not materially alter its behaviour under smoothly varying objectives. On the piecewise-preference-shift tasks (where tasks recur every three time steps), CPPO outperforms PPO in later training epochs—consistent with its improved retention design—but suffers drops on repeated tasks and matches PPO on the test set, revealing a limited capacity to generalize retention across recurrences. Finally, in the CPPO-RL dataset, CPPO adapts more rapidly at test time, mirroring the Lipschitz results and highlighting that this synthetic benchmark replicates the smooth continuity patterns of our Lipschitz setup.

These findings demonstrate the necessity for richer, more varied benchmarks when evaluating RL-based fine-tuning methods for large language models and algorithms that address the stability-plasticity tradeoff more robustly. The pre-existing dataset used by Zhang et al. (2024b) captures only a narrow slice of non-stationarity. A broader suite of continuity patterns is required to reveal where algorithms succeed or fail in balancing adaptability against forgetting, provided by our work and AIF-GEN. In sum, while CPPO consistently attains superior training-time performance, its test-time gains diminish under shifts in dataset type and non-stationarity; it remains prone to forgetting specific tasks despite its adaptability enhancements. Due to its assumption of stationary preferences, CPPO regresses on our datasets where the preferences change.

C.3 ADDITIONAL DPO RESULTS

Figure 9 shows DPO performance, analogous to Figure from the main paper, but on the CPPO RL dataset. We observe similar patterns to those in the long-piecewise dataset: high accuracy on training samples with clear drops at task transitions. Test accuracy steadily improves, indicating that DPO can generalize across task switches in this setting. In contrast to its behaviour on our other datasets, where generalization was limited, this result highlights how different forms of non-stationarity, simulated using `AIF-GEN`, can stress distinct dimensions of continual learning in RLHF.

To estimate forgetting, we compute the normalized backward transfer as $\frac{R_{k,T}-R_{k,\text{last}}}{R_{k,\text{last}}}$, where $R_{k,\text{last}}$ is the performance of the last checkpoint trained on task k, evaluated on task k, and $R_{k,T}$ is the performance of a later checkpoint $T >$ last, also evaluated on task k. In Fig. 10, we plot the mean and max/min values of normalized backward transfer across all previously encountered datasets as a function of time t.

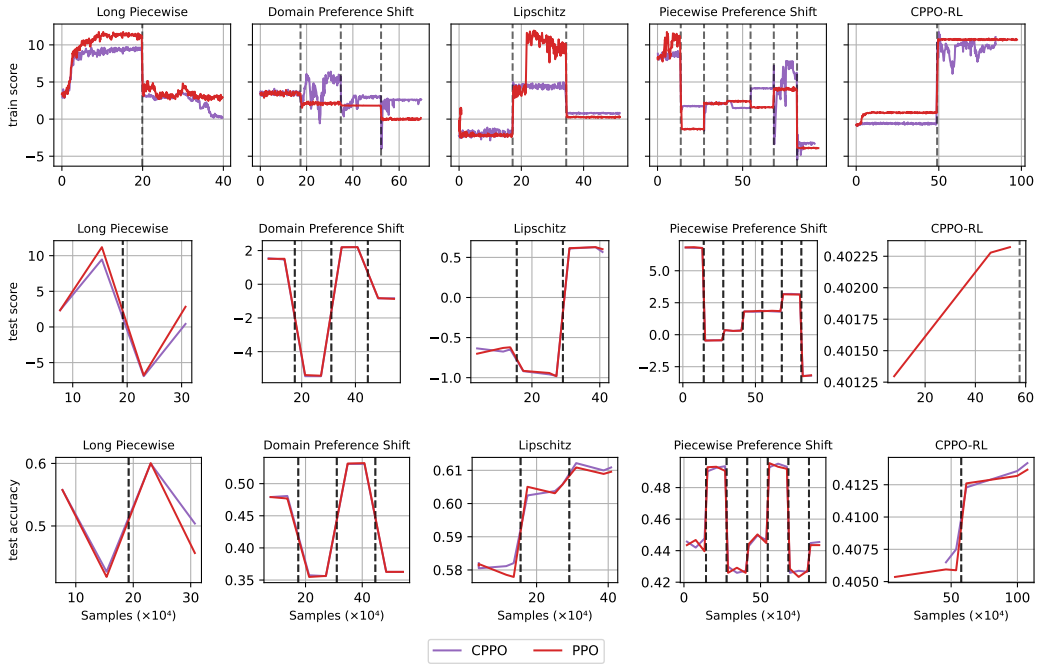


Figure 8: Training and test scores for PPO and CPPO, along with test accuracy during training. Dashed lines indicate task switches in the continual learning datasets. Train and test scores are computed based on the reward model using the training and test datasets, respectively. Test accuracy is computed similarly to DPO, using the implicit reward model.

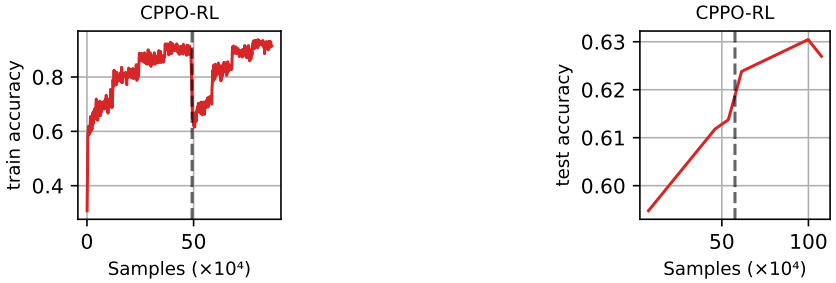


Figure 9: Train and test accuracy for DPO on CPPO dataset using the implicit reward model. Dashed lines indicate a task switch.

We observe that, on average, DPO exhibits a small degree of forgetting. However, in some tasks, positive transfer also occurred. For example, the Piecewise Preference Shift dataset shows maximum values above zero, indicating positive transfer for specific tasks. This is expected, given the dataset's cyclical repetition of three subtasks.

C.4 AIF-GEN DATASETS STATISTICS

In this appendix, we provide detailed statistics for the synthetic datasets generated using `AIF-GEN`, complementing the quality analysis presented in Figure 5 of the main paper. Tables 7, 8, and 9 break down sample counts, prompt entropy, response entropy (chosen and rejected), and coherence scores across individual datasets categorized by objective, preference, and domain. While each dataset was designed to contain 10,000 examples, the final counts are slightly lower due to filtering steps that excluded samples affected by API failures (e.g., VLLM or OpenAI), token limit violations, or parsing

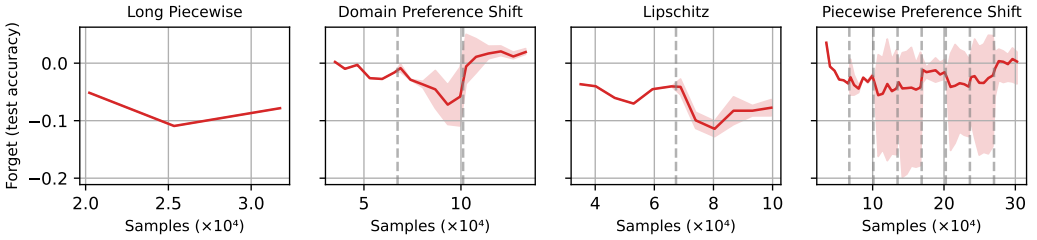


Figure 10: Mean normalized backward transfer for DPO, computed using test accuracy across all previous datasets and plotted over training time. Shaded area indicates max/min values.

errors during structured binding. These tables offer a granular view of the data diversity and quality underpinning our lifelong RLHF benchmarks.

Table 7: Validation statistics for Generate tasks.

	Politics		
Statistic	**Formal**	**Rapper**	**Shakespeare**
Sample Count	9992	9985	9975
Prompt Entropy	6.977	6.977	6.980
Chosen Entropy	7.353	7.583	7.701
Rejected Entropy	7.424	7.590	7.617
Coherence Chosen	8.785	8.616	8.606
Coherence Rejected	8.744	8.632	8.612

As expected, we observe greater variation across datasets defined by different objectives, reflecting the diversity of generation tasks in `AIF-GEN`. Interestingly, coherence and entropy also vary slightly across stylistic preferences. For example, in generation tasks, responses generated with the rapper style exhibit lower coherence scores (8.616 and 8.632) compared to the formal style (8.785 and 8.744), while also showing higher token entropy—suggesting broader vocabulary usage and greater linguistic variability. Similarly, rapper and Shakespeare preferences tend to produce responses with more lexical diversity. In summarization tasks, the expert preference consistently yields a higher coherence score than its eli5 counterpart across domains, a trend also observed in Q&A datasets. Notably, hinted and direct preferences yield nearly identical coherence metrics, indicating that `AIF-GEN` maintains consistent quality across subtly different instructional styles.

Table 8: Validation statistics for Summary tasks.

	Education		**Politics**		**Tech**		**Physics Eli5**		**Physics Expert**		**Physics Highschool**	
Statistic	Eli5	Expert	Eli5	Expert	Eli5	Expert	Eli5	Expert	Eli5	Expert	Eli5	Expert
Sample Count	9996	9995	9996	9995	9996	9995	9997	9997	9999	9999	9996	9996
Prompt Entropy	7.121	7.124	6.938	7.340	7.012	6.732	7.014	7.014	7.012	7.340	7.012	7.014
Chosen Entropy	7.411	7.440	7.297	7.319	7.319	7.174	7.297	7.297	7.297	7.319	7.319	7.297
Rejected Entropy	7.448	7.478	7.329	7.388	7.362	7.249	7.351	7.351	7.362	7.388	7.362	7.351
Coherence Chosen	8.864	8.983	8.543	8.574	8.643	8.889	8.866	8.866	8.643	8.574	8.643	8.866
Coherence Rejected	8.944	8.972	8.640	8.659	8.792	8.870	8.888	8.888	8.792	8.659	8.792	8.888

C.5 Empirical proof for AIF-Gen Non-stationarities

Let $\theta_i \in \mathbb{R}^d$ be the parameters of reward model M_i for $i = 1, 2, 3$, and define the "task vectors" $T_{12} = \theta_2 - \theta_1$ and $T_{23} = \theta_3 - \theta_2$. For $\alpha \in [0, 1]$, set $\theta(\alpha) = \theta_1 + \alpha\, T_{12}$, we load these weights into a copy of M_1, and compute the reward-score histogram $R(\alpha)$ over our fixed 10 000 prompts. We then calculate the adjacent Wasserstein distances $W\big(R(\alpha_i), R(\alpha_{i+1})\big)$, normalize each by $(\alpha_{i+1} - \alpha_i)\,\|T_{12}\|$, and take their maximum to obtain an empirical Lipschitz constant K_{12}; repeating along T_{23} yields K_{23}. On the "AIF-Gen Lipschitz" dataset we measure $\max(K_{12}, K_{23}) \approx 5$, whereas on the "AIF-Gen Piecewise Preference Shift" dataset it is ≈ 10, showing that the piecewise dataset induces larger, less smooth jumps in the learned reward distribution therefore making it a more difficult task to learn.

Table 9: Validation statistics for Q&A tasks.

	Education				Politics		Tech	
	Direct	**Eli5**	**Expert**	**Hinted**	**Eli5**	**Expert**	**Healthcare Eli5**	**Healthcare Expert**
Sample Count	9996	9991	9996	9991	9977	9982	9997	9991
Prompt Entropy	6.166	6.154	6.149	6.158	5.614	5.606	5.627	5.613
Chosen Entropy	7.620	7.584	7.755	7.539	7.329	7.528	7.456	7.626
Rejected Entropy	7.693	7.596	7.688	7.565	7.325	7.439	7.460	7.527
Coherence Chosen	8.995	8.827	9.046	8.837	8.642	8.828	8.846	9.057
Coherence Rejected	8.916	8.845	9.007	8.937	8.672	8.776	8.861	9.017

D AIF-GEN COMMAND-LINE INTERFACE (CLI)

AIF-Gen is primarily meant to be used as a command-line tool when generating and manipulating synthetic continual RLHF datasets. The tool is invoked using:

```
$ aif --help
```

Available commands:

- `generate` – Generate a new `ContinualAlignmentDataset`.
- `merge` – Interactively merge multiple datasets.
- `preview` – Interactively preview dataset samples.
- `sample` – Downsample datasets by ratio or count.
- `transform` – Apply dataset transformations.
- `validate` – Run dataset validation checks.

For usage examples, refer to: https://aif-gen.readthedocs.io/en/latest/cli

For installation instructions, please consult: https://aif-gen.readthedocs.io/en/latest

GLOBAL OPTIONS

`-log_file FILE`	Optional log file path (default: `aif_gen.log`)
`-help`	Show help message and exit

GENERATE

Generates a new continual dataset using a vLLM-compatible model.

`-data_config_name`	Path to the dataset configuration file
`-model`	Name of vLLM model for generation
`-output_file`	Output path for the generated dataset
`-random_seed`	Random seed for reproducibility (default: `0`)
`-dry_run`	Simulate generation a dry run (default: `False`)
`-temperature`	LLM Sampling temperature (default: `0.99`)
`-hf_repo_id`	(Optional) Save to Hugging Face repository
`-max_tokens_prompt_response`	Token limit for prompts (default: `1024`)
`-max_tokens_chosen_rejected_response`	Token limit for responses (default: `2048`)
`-max_concurrency`	Max number of concurrent inference requests to send to the vLLM server (default: `256`)

MERGE

Interactively merges multiple datasets via terminal prompts.

(No additional flags; operates interactively.)

PREVIEW

Preview a dataset interactively by cycling through examples.

`-input_data_file`	Path to the input dataset
`-shuffle`	Whether to shuffle samples before display (default: `True`)
`-hf_repo_id`	(Optional) Load from Hugging Face repository

SAMPLE

Downsample a dataset by ratio or absolute sample count.

`-input_data_file`	Path to the input dataset
`-keep_ratio_train`	Fraction of training data to retain
`-keep_ratio_test`	Fraction of test data to retain
`-output_data_file`	Path to write the transformed dataset
`-random_seed`	Seed for reproducibility (default: `0`)
`-keep_amount_train`	(Optional) Absolute number of training samples to retain
`-keep_amount_test`	(Optional) Absolute number of test samples to retain
`-hf_repo_id`	(Optional) Load from Hugging Face repository
`-hf_repo_id_out`	(Optional) Save to Hugging Face repository

TRANSFORM

Transform a `ContinualAlignmentDataset`.

preference_swap Swap 'chosen' and 'rejected' responses probabilistically.

`-input_data_file`	Path to the input dataset
`-output_data_file`	Path to write the transformed dataset
`-p`	Swap probability (default: `1`)
`-random_seed`	Seed for reproducibility (default: `0`)
`-hf_repo_id`	(Optional) Load from Hugging Face repository
`-hf_repo_id_out`	(Optional) Save to Hugging Face repository

split Split dataset into train and test partitions.

`-input_data_file`	Path to the input dataset
`-output_data_file`	Path to write the transformed dataset
`-test_sample_ratio`	Ratio for the test split (default: `0.15`)
`-random_seed`	Seed for reproducibility (default: `0`)
`-hf_repo_id`	(Optional) Load from Hugging Face repository
`-hf_repo_id_out`	(Optional) Save to Hugging Face repository

VALIDATE

Run dataset validation with several configurable checks.

`-input_data_file`	Path to the input dataset
`-output_data_file`	Path to write the validation results
`-validate_count`	Enable count-based checks
`-validate_entropy`	Enable entropy-based evaluation
`-validate_llm_judge`	Enable LLM judgment scoring
`-validate_embedding_diversity`	Enable embedding diversity checks
`-model`	LLM model name for judgment
`-embedding_model`	Embedding model name
`-embedding_batch_size`	Batch size for embedding calculation (default: `256`)
`-max_tokens_judge_response`	Token limit for LLM judgment response (default: `128`)
`-random_seed`	Random seed for reproducibility (default: `0`)
`-dry_run`	Simulate LLM judge with a dry run (default: `False`)
`-hf_repo_id`	(Optional) Load dataset from Hugging Face repository
`-max_concurrency`	Max number of concurrent inference requests to send to the vLLM server (default: `256`)

E PROMPT TEMPLATES

In this section, we provide the templates with which AIF-Gen datasets were created. As described in the main text, the framework utilizes a prompt and response mapper internally for the generation task given external data generation configuration YAML files provided by the user; which can all be found in the GitHub repository. Appendix E.1, E.2, and E.3 display the prompts used respectively.

E.1 PROMPT MAPPER

The following prompt template is used to generate task-specific prompts for our alignment tasks:

```
Generate a text that fulfills the objective below.
Do exactly what the objective says: [OBJECTIVE].
The description must include the following seed words: [SEED_WORDS].
Do not include any meta commentary, instructions, or extra text
(e.g., avoid phrases like "User asks" or additional context).
The output should be clear and self-contained.
You don't need to start by saying "prompt:".
Ensure that the generated response adheres to ethical practices,
avoids biases, and respects the target audience's needs.
```

where `[OBJECTIVE]` is the specific alignment task objective, and `[SEED_WORDS]` are domain-specific terms sampled from task components to contextualize the generation.

E.2 RESPONSE MAPPER

```
Generate a 'chosen' and 'rejected' response pair to the following
prompt: [TASK_PROMPT].
The 'chosen' response should respond to the prompt according to the
following preference: [PREFERENCE].
The 'rejected' response should still respond to the prompt according
to the preference but negligibly worse in its quality, however still
close to the chosen response so it confuses the reader which one is
actually better.
Consider exactly the same style and lengths for the chosen and
rejected please.
You don't need to start your response by saying "here is the response"
nor to give any meta-explanation. Just provide the response.
```

where `[TASK_PROMPT]` is the previously generated task prompt, and `[PREFERENCE]` represents the specific preference criteria from the alignment task.

E.3 Data Generation Configurations

The configurations provided below have the following format:

1. Objective
2. Preference
3. Domain

education_qna_direct

Ask a non-trivial math (you choose randomly what topic in math) or social sciences or physics or chemistry question. Your response is merely a literal question and nothing else please.

Directly answer the question.

`education`

education_qna_eli5

Ask a non-trivial math (you choose randomly what topic in math) or social sciences or physics or chemistry question. The prompt is merely a literal question and nothing else please.

Explain the answer to the question at a level that could be understood by a five year old.

`education`

education_qna_expert

Ask a non-trivial math (you choose randomly what topic in math) or social sciences or physics or chemistry question. The prompt is merely a literal question and nothing else please.

Explain the answer to the question at an expert level. Draw from technical literature when necessary, and add complex examples to further support the student learning.

`education`

education_qna_hinted

Ask a non-trivial math (you choose randomly what topic in math) or social sciences or physics or chemistry question. The prompt is merely a literal question and nothing else please.

Do not directly reveal the answer to the question. Instead, guide the student with a relevant hint.

`education`

education_summary_eli5

Generate an article on the topic of math or social sciences or physics or chemistry (you choose the exact detailed topic) that would be written by a professional expert in the field.

Summarize the body of text at a level that could be understood by a five year old.

`education`

education_summary_expert

Generate an article on the topic of math or social sciences or physics or chemistry (you choose the exact detailed topic) that would be written by a professional expert in the field.

Summarize the body of text at an expert level without making the summary too long or too short. Draw from technical literature when necessary.

`education`

politics_generate_formal

Generate a body of text on a political topic (you choose randomly what topic in politics) that would be found in a blog article.

Continue the story but in a formal style.

`politics`

politics_generate_rapper

Generate a body of text on a political topic (you choose randomly what topic in politics) that would be found in a blog article.

Continue the story but in the style of a rapper.

`politics`

politics_generate_shakespeare

Generate a body of text on a political topic (you choose randomly what topic in politics) that would be found in a blog article.

Continue the story but in the style of Shakespeare.

`politics`

politics_qna_expert

Ask a political (you choose randomly what topic in politics) question. The prompt is merely a literal question and nothing else please.

Explain the answer to the question at an expert level. Draw from technical literature when necessary, and add complex examples to further support the answer.

`politics`

politics_summary_eli5

Generate a body of text on the topic of politics (you choose randomly what topic in politics) that would be found in an article written by an expert in the field.

Summarize the body of text at a level that could be understood by a five year old.

`politics`

politics_summary_expert

Generate a body of text on the topic of politics(you choose randomly what topic in politics) that would be found in an article written by an expert in the field.

Summarize the body of text at an expert level. Draw from technical literature when necessary.

`politics`

tech_healthcare_qna_eli5

Ask a healthcare (you choose randomly what topic in health sciences) or technology related (you choose randomly what topic related to technology) question. The prompt is merely a literal question and nothing else please.

Explain the answer to the question at a level that could be understood by a five year old.

`Technology and Healthcare`

tech_healthcare_qna_expert

Ask a healthcare (you choose randomly what topic in health sciences) or technology related (you choose randomly what topic related to technology) question. The prompt is merely a literal question and nothing else please.

Explain the answer to the question at an expert level. Draw from technical literature when necessary, and add complex examples to further support the answer.

`Technology and Healthcare`

tech_physics_summary_eli5

Generate an article on the topic of healthcare (you choose the exact detailed topic in health sciences) or technology (you choose randomly what topic related to technology) that would be written by a professor or a pioneering expert in the field.

Summarize the body of text at a level that could be understood by a five year old.

`Technology and Physics`

tech_physics_summary_expert

Generate a body of text on the topic of healthcare (you choose randomly what topic in health sciences) or technology (you choose randomly what topic related to technology) that would be found in a blog article written by an expert in the field.

Summarize the body of text at an expert level. Draw from technical literature when necessary.

`Technology and Physics`

tech_physics_summary_highschool

Generate an article on the topic of healthcare (you choose the exact detailed topic in health sciences) or technology (you choose randomly what topic related to technology) that would be written by a professor or a pioneering expert in the field.

Summarize the body of text at a level that could be understood by a regular high school student.

`Technology and Physics`