AIF-GEN: Open-Source Platform and Synthetic Dataset Suite for Reinforcement Learning on Large Language Models

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

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. 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 and evaluate its quality using an LLM. We also perform human evaluation on a subset of the generated datasets to further confirm its quality. Our results show AIF-GEN's potential to support the development of traditional and lifelong RLHF algorithms that align LLMs.

Code: ComplexData-MILA/AIF-Gen

Data: 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 its adaptability in dynamic settings (Jurenka et al., 2024). Further, large organizations carefully curate private datasets for alignment, which makes it more challenging for independent researchers to reproduce and benchmark results on diverse datasets.

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 and Chen, 2023; Zhang et al., 2023).

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. These limitation hinders the progress in alignment research, especially lifelong alignment where models must adapt to distributional drifts.

In this work, we introduce AIF-GEN—the first platform for scalable synthetic data generation for both traditional and lifelong RLHF. By parameterizing user-defined objectives, domains, and preferences, AIF-GEN enables systematic creation of diverse datasets to support fine-tuning and continual alignment of LLMs. It combines open-source accessibility, flexible synthetic data generation, and validation features with native support for non-stationary prompts and preferences (see 1). Using the platform, we instantiate 18 synthetic preference datasets totalling roughly 170,000 prompts and 340,000 preference annotations. Designed to be LLM-agnostic, scalable, and customizable, AIF-GEN lowers the barrier to entry for traditional and lifelong RLHF research and provides a foundation for advancing reproducibility in adaptive, preference-aligned language models.

Summary of contributions:

1. We introduce **AIF-GEN**, the first open-sourced synthetic data generation platform tailored to traditional and lifelong RLHF (§ 3);

[*]Equal contribution [1]Mila - Quebec AI Institute [2]School of Computer Science, McGill University [3]DIRO, Université de Montréal [4]Vector Institute [5]CIFAR AI Chair. Correspondence to: Shahrad Mohammadzadeh <shahrad.mohammadzadeh@mila.quebec>, Jacob Chmura <jacob.chmura@mail.mcgill.ca>.

Proceedings of the ICML 2025 Workshop on Championing Open-source Development in Machine Learning (CODEML '25).

2. Using AIF-GEN, we generate a diverse suite of synthetic datasets that capture varying types and degrees of non-stationarity to support controlled experiments(§ 3.3);

3. We validate the quality of our synthetic datasets using LLM and human evaluations(§ 4.1);

2. Background

The RLHF process begins by fine-tuning a base language model, π^0, via supervised learning to obtain an SFT model, π^{SFT}. This model is then prompted with inputs x to produce two responses, (y_1, y_2), which are evaluated—either by humans or an automated judge (e.g., in RLAIF). One response is marked as preferred (e.g., y_1) and the other as rejected (e.g., y_2), denoted as $y_1 \succ y_2 \mid x$. These preferences are assumed to reflect an underlying reward function $r^*(y, x)$, often modelled using the Bradley-Terry framework (Bradley and Terry, 1952). Alternative models include Nash (Munos et al., 2024; Zhu et al., 2024) and Plackett-Luce (Plackett, 1975). From these comparisons, a preference dataset is constructed: $D = \{x^i, y_c^i, y_r^i\}_{i=1}^N$, where y_c^i and y_r^i denote the chosen and rejected responses for prompt x^i. A reward model r_ϕ is trained on this dataset, serving as the training signal for reinforcement learning. The RL objective fine-tunes π^{SFT} to maximize reward while penalizing divergence from the original SFT distribution:

$$\max_{\pi_\theta} \mathbb{E}_{x\sim D,\, y\sim\pi_\theta(\cdot|x)} \Big[r_\phi(y, x) - \beta\, \mathbb{D}_{KL}\left(\pi_\theta(\cdot|x) \,\|\, \pi^{SFT}(\cdot|x)\right) \Big] \quad (1)$$

where β balances reward maximization with alignment to π^{SFT}. While various RL algorithms can be applied, PPO(Schulman et al., 2017) remains the most commonly used in practice.

2.1. Lifelong RLHF

In lifelong RLHF, the objective is to align the model using the latest batch of preference data that reflects the current preferences, while retaining knowledge about the past that could be useful in future (Zhang et al., 2024b;a; Wu et al., 2024). The preference data distribution changes when either the prompt distribution or the preference distribution for the two responses change over time.

3. AIF-GEN Platform

Human annotation for traditional and lifelong RLHF is costly and slow. To scale alignment research, and to enable standardization while reporting results, we introduce AIF-GEN, a platform for generating synthetic preference data across diverse topics for traditional and lifelong RLHF (§A). Users define sequences of *Alignment Tasks* in structured YAML files, specifying evolving objectives, domains, and preferences (§3.1). These feed into an asynchronous vLLM-powered engine that produces prompt–response pairs at scale (Kwon et al., 2023). Prompt templates are managed by the *Prompt Mapper*, while the *Response Mapper* synthesizes candidate completions and formats the final preference samples (§3.2). Our platform also supports validation, transformation, and dataset publishing via CLI.

3.1. 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. This decomposition allows researchers to isolate and manipulate distinct modes of drift:

Objective: the objective is, in principle, fully customizable: users can define any task that fits their application. However, in practice, most use cases cluster around a few common categories like question answering, summarization, and text generation. To support these, we provide template datasets for each as open-source examples, which users can readily adapt to their specific problems.

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".

3.2. Mapper Internals

Prompt Mapper internals are shown in Figure 2a. 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, engineering). Seed words are combined with the task objective (e.g., Q&A) to generate a Meta Prompt, which is passed to the LLM.

Response Mapper internals are shown in Figure 2b. Each prompt (e.g., a biology question) is paired with auxiliary styles (e.g., length, humour) to generate diverse responses.

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.

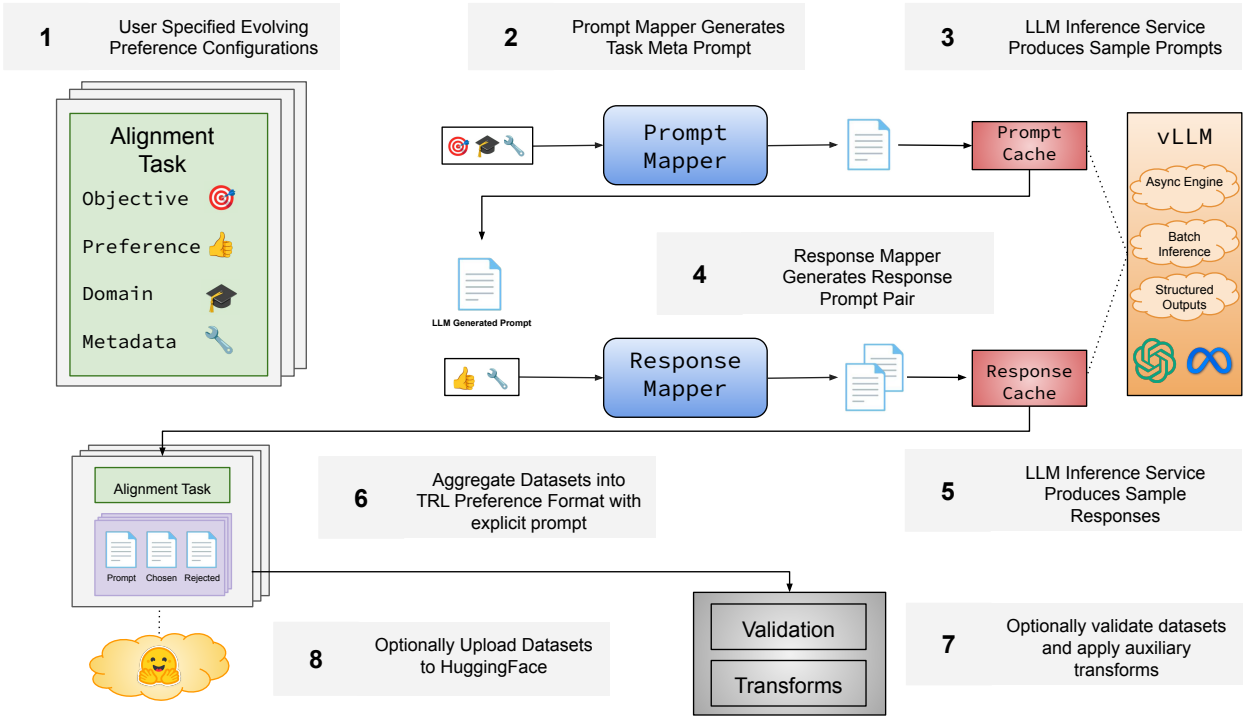


Figure 1: High-level AIF-GEN design. (1) Users specify RLHF tasks (objectives, preferences, domains) and metadata in YAML configs. (2) Prompt Mapper generates meta-prompts, which (3) produce sample prompts using an LLM. (4) Response Mapper pairs these with preferences to create response prompts, which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

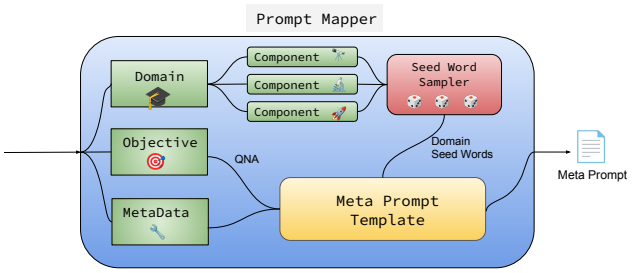
The task preference (e.g., "explain like I'm five") is probabilistically included (configurable) to influence the Judge Prompt. This helps calibrate the difficulty of distinguishing preferred responses. Resulting prompts are sent to the inference engine to produce chosen/rejected pairs.

3.3. Datasets

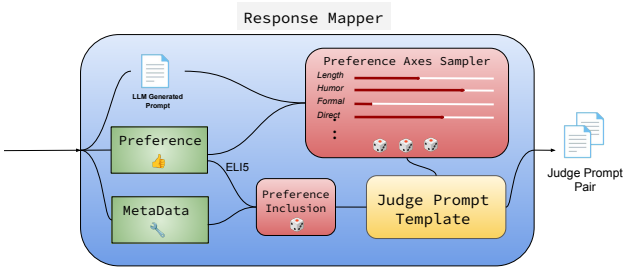
We used AIF-GEN with GPT-4o-mini to generate our data with a temperature of 0.99, a maximum prompt length of 1024 tokens, and a response cap of 2048 tokens.

Static Datasets. We generated static datasets—each defined by an objective (e.g., Q&A, summarization, text generation), domain (e.g., education, politics, tech/healthcare, tech/-physics), and stylistic preference (e.g., ELI5, expert, formal, Shakespearean). Prompts were created using domain-specific seed vocabularies (two seeds per prompt), and responses were sampled in 3 styles. Each dataset contains 10,000 examples; full templates and preference configurations are provided in the appendix.

Continual Datasets. We built four continual datasets to study alignment under drift by merging static subsets and simulating structured transitions in preference, domain, and objective. The Lipschitz dataset gradually increases preference complexity within tech/physics summarization: *ELI5* → *high school* → *expert*. Piecewise Preference cycles through *rapper*, *Shakespearean*, and *formal* styles in political generation, repeated over three cycles. Piecewise Q&A alternates between *hinted* and *directed* formats (5k samples each) to vary non-stationarity frequency. The most



(a) *Prompt Mapper* internals.



(b) *Response Mapper* internals.

Figure 2: (a) Domains are decomposed into semantic components and sampled via seed word vocabularies, combined with objectives using meta-prompt templates. Metadata supports custom prompt formatting. (b) Preferences and prompts are passed to a judge template, with optional auxiliary styles to vary response subtlety and difficulty. Metadata controls preference strength for reward modelling tasks.

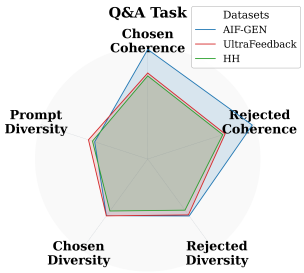
complex dataset combines shifts across all axes: education Q&A (*ELI5*), education Q&A (*expert*), political summarization (*ELI5*), and tech/healthcare Q&A (*expert*), serving as a comprehensive benchmark for lifelong RLHF.

4. Experiments

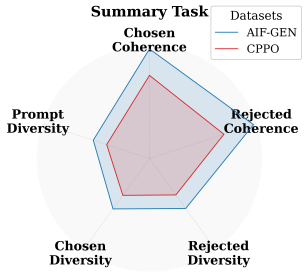
We validate the quality of generated data using an LLM judge and humans.

4.1. LLM Validation

Figure 3 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 3: `AIF-GEN` generates higher quality RLHF datasets for both Q&A and summarization tasks compared to previous datasets.

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

4.2. Human Evaluation

We conducted a targeted evaluation on education Q&A samples using three stylistic preferences—Hinted, ELI5, and Expert—chosen for their varying alignment difficulty. For each, we sampled 50 prompt–response pairs and asked three annotators to select the better-aligned response or mark *both*, *neither*, and flag incoherent samples.

Table 2 reports four metrics: Unanimous Consensus Rate, Fleiss' Kappa, LLM–Human Agreement, and Inter-Human Agreement, with confidence intervals from 1,000 bootstraps. Only 5 of 450 samples were flagged as incoherent. We observe solid agreement (48–64% consensus, Kappa ≈ 0.5 for ELI5/Expert) and LLM judge accuracy near 60%, consistent with prior work (Cui et al., 2023). Lower scores for Hinted reflect its subtler alignment. In Appendix B.2, we plot a classification heatmap and demonstrate that LLMs align well with human judgments, with ambiguity concentrated in *both*/*neither* cases—especially for ELI5, which may benefit from stronger preference conditioning in generation.

4.3. Empirical proof for AIF-Gen Non-stationarities

To better understand how AIF-Gen datasets induce shifts in the learned reward models, we analyze the sensitivity of reward distributions to changes in model parameters across tasks. This provides an approximate characterization of the degree of non-stationarity present in different dataset variants and highlights how abrupt or smooth these shifts

are in parameter space. While these are simple heuristic measurements, they offer a practical and interpretable way to compare dataset difficulty in settings where precise quantification is not feasible.

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 $T_{23} = \theta_3 - \theta_2$ which capture the parameter-space trajectories between successive tasks. For $\alpha \in [0, 1]$, we interpolate along these trajectories by setting $\theta(\alpha) = \theta_1 + \alpha T_{12}$, loading these weights into a copy of M_1, and compute the reward-score histogram $R(\alpha)$ over our fixed 10 000 prompts. This procedure gives a smooth parameterized path through model space, allowing us to monitor how the output distribution evolves under controlled perturbations.

To quantify local sensitivity along this path, we compute 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 estimate an empirical Lipschitz constant K_{12}; repeating along T_{23} yields K_{23}. Intuitively, this constant measures how sharply the reward distribution changes as we move in parameter space. On the "AIF-Gen Lipschitz" dataset we measure $\max(K_{12}, K_{23}) \approx 5$, indicating relatively smooth transitions between tasks. In contrast, on the "AIF-Gen Piecewise Preference Shift" dataset, we observe an empirical lipschitaz estimate of ≈ 10. This indicates that the piecewise dataset induces larger, more abrupt shifts in the learned reward distributions, with sharper discontinuities in model behavior across tasks.

5. Discussion, Limitations, and Conclusion

The success of RL has been driven by open-source simulators like Atari (Bellemare et al., 2013) and MuJoCo (Todorov et al., 2012), which enabled rapid progress through standardized, reproducible experimentation. In contrast, traditional and lifelong RLHF research is bottlenecked by the absence of scalable, time-evolving human preference data. To address this, we introduce `AIF-GEN`—a synthetic preference generation platform that plays a similar role for alignment research, enabling controlled studies of dynamic tasks and shifting user preferences.

While powerful, `AIF-GEN` makes several design trade-offs. Our prompt templates are handcrafted but easily extensible, allowing users to define new domains and alignment styles. We focus on GPT-4o-mini for data generation, balancing quality and cost; however, the platform is model-agnostic and supports future integration of emerging models. Automated evaluations use LLM judges that may carry bias, but targeted human assessments validate overall quality. As a platform—not a fixed benchmark—`AIF-GEN`'s adaptability ensures its relevance across evolving research objectives.

Rooted in open-source principles, `AIF-GEN` is designed to grow with the community. We invite researchers and practitioners to contribute tasks and suggest new forms of non-stationarity, building a shared ecosystem for studying alignment under distribution shift. In doing so, we hope to establish `AIF-GEN` as a foundational platform for reproducible, extensible research in lifelong RLHF.

Impact Statement

The research presented in this work aims to significantly advance alignment methodologies in reinforcement learning from human (and AI) feedback, especially in dynamic, evolving settings. AIF-GEN offers the first open-source, LLM-agnostic platform for generating large-scale synthetic preference datasets tailored for both traditional and lifelong RLHF. By enabling fine-grained control over objectives, domains, and user preferences—including non-stationary drift—this platform facilitates reproducible experimentation in alignment, a current bottleneck in the field.

The synthetic datasets and continual learning scenarios generated with AIF-GEN can democratize access to high-quality alignment resources, lowering the barrier for researchers and practitioners outside of large tech organizations. As alignment research becomes increasingly vital for deploying safe and responsible AI systems, especially LLMs, tools like AIF-GEN may help establish standardized benchmarks and promote robust model evaluation across shifting user needs.

We acknowledge the broader societal implications of LLM alignment, particularly as models are deployed in sensitive areas such as education, healthcare, and policy communication. While AIF-GEN reduces reliance on static human feedback, care must be taken to ensure synthetic preferences reflect diverse and representative values. We encourage community-driven governance, transparency in dataset creation, and continual validation (including human oversight) to mitigate potential misalignment or biases introduced by automation in feedback generation.

Overall, AIF-GEN provides infrastructure to accelerate research in dynamic alignment settings, helping move the field toward safer, more adaptive AI systems.

Acknowledgements

Funding support for project activities has been partially provided by the Canada CIFAR AI Chair. We also express our gratitude to Compute Canada, Mila, and Oak Ridge National Laboratory compute clusters for their support in providing facilities for our evaluations.

References

Abel, D., Barreto, A., Roy, B. V., Precup, D., van Hasselt, H., and Singh, S. (2023). A definition of continual reinforcement learning.

Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., Chen, A., Goldie, A., Mirhoseini, A., McKinnon, C., Chen, C., Olsson, C., Olah, C., Hernandez, D., Drain, D., Ganguli, D., Li, D., Tran-Johnson, E., Perez, E., Kerr, J., Mueller, J., Ladish, J., Landau, J., Ndousse, K., Lukosuite, K., Lovitt, L., Sellitto, M., Elhage, N., Schiefer, N., Mercado, N., DasSarma, N., Lasenby, R., Larson, R., Ringer, S., Johnston, S., Kravec, S., Showk, S. E., Fort, S., Lanham, T., Telleen-Lawton, T., Conerly, T., Henighan, T., Hume, T., Bowman, S. R., Hatfield-Dodds, Z., Mann, B., Amodei, D., Joseph, N., McCandlish, S., Brown, T., and Kaplan, J. (2022). Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv: 2212.08073*.

Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. (2013). The arcade learning environment: An evaluation platform for general agents. *Journal of artificial intelligence research*, 47:253–279.

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

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. (2017). Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4299–4307.

Cui, G., Yuan, L., Ding, N., Yao, G., Zhu, W., Ni, Y., Xie, G., Liu, Z., and Sun, M. (2023). Ultrafeedback: Boosting language models with high-quality feedback.

Dai, J., Pan, X., Sun, R., Ji, J., Xu, X., Liu, M., Wang, Y., and Yang, Y. (2023). Safe rlhf: Safe reinforcement learning from human feedback.

Jurenka, I., Kunesch, M., McKee, K. R., Gillick, D., Zhu, S., Wiltberger, S., Phal, S. M., Hermann, K., Kasenberg, D., Bhoopchand, A., Anand, A., Pîslar, M., Chan, S., Wang, L., She, J., Mahmoudieh, P., Rysbek, A., Ko, W.-J., Huber, A., Wiltshire, B., Elidan, G., Rabin, R., Rubinovitz, J., Pitaru, A., McAllister, M., Wilkowski, J., Choi, D., Engelberg, R., Hackmon, L., Levin, A., Griffin, R., Sears, M., Bar, F., Mesar, M., Jabbour, M., Chaudhry, A., Cohan, J., Thiagarajan, S., Levine, N., Brown, B., Gorur, D., Grant, S., Hashimshoni, R., Weidinger, L., Hu, J., Chen, D., Dolecki, K., Akbulut, C., Bileschi, M., Culp, L., Dong, W.-X., Marchal, N., Deman, K. V., Misra, H. B., Duah, M., Ambar, M., Caciularu, A., Lefdal, S., Summerfield, C., An, J., Kamienny, P.-A., Mohdi, A., Strinopoulous, T., Hale, A., Anderson, W., Cobo, L. C., Efron, N., Ananda, M., Mohamed, S., Heymans, M., Ghahramani, Z., Matias, Y., Gomes, B., and Ibrahim, L. (2024). Towards responsible development of generative ai for education: An evaluation-driven approach.

Khetarpal, K., Riemer, M., Rish, I., and Precup, D. (2022). Towards continual reinforcement learning: A review and perspectives. *Journal of Artificial Intelligence Research*, 75:1401–1476.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., and Stoica, I. (2023). Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Li, X. and Chen, Y. (2023). Reinforcement learning from ai feedback. *arXiv preprint arXiv:2305.12345*.

Marten, R., Vu, T., Ji, C. C.-J., Sharma, K., Pimpalgaonkar, S., Dimakis, A., and Sathiamoorthy, M. (2025). Curator: A tool for synthetic data creation. https://github.com/bespokelabsai/curator.

Meng, R., Liu, Y., Joty, S. R., Xiong, C., Zhou, Y., and Yavuz, S. (2024). Sfr-embedding-mistral:enhance text retrieval with transfer learning. Salesforce AI Research Blog.

Munos, R., Valko, M., Calandriello, D., Azar, M. G., Rowland, M., Guo, Z. D., Tang, Y., Geist, M., Mesnard, T., Michi, A., Selvi, M., Girgin, S., Momchev, N., Bachem, O., Mankowitz, D. J., Precup, D., and Piot, B. (2024). Nash learning from human feedback.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., and et al. (2022). Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 27730–27744.

Patel, A., Raffel, C., and Callison-Burch, C. (2024). DataDreamer: A tool for synthetic data generation and reproducible LLM workflows. In Ku, L.-W., Martins, A., and Srikumar, V., editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3781–3799, Bangkok, Thailand. Association for Computational Linguistics.

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

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. In *Advances in Neural Information Processing Systems*, volume 30, pages 1133–1143.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D. M., Lowe, R., and et al. (2020). Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3008–3021.

Sun, Z., Shen, S., Cao, S., Liu, H., Li, C., Shen, Y., Gan, C., Gui, L.-Y., Wang, Y.-X., Yang, Y., Keutzer, K., and Darrell, T. (2023). Aligning large multimodal models with factually augmented rlhf.

Todorov, E., Erez, T., and Tassa, Y. (2012). Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE.

Wu, T., Luo, L., Li, Y.-F., Pan, S., Vu, T.-T., and Haffari, G. (2024). Continual learning for large language models: A survey. *arXiv preprint arXiv:2402.01364*.

Zhang, H., Gui, L., Lei, Y., Zhai, Y., and et al. (2024a). Copr: Continual human preference learning via optimal policy regularization. *arXiv preprint arXiv:2402.14228*.

Zhang, H., Lei, Y., Gui, L., Yang, M., He, Y., Wang, H., and Xu, R. (2024b). CPPO: Continual learning for reinforcement learning with human feedback. In *The Twelfth International Conference on Learning Representations*.

Zhang, Q., Li, X., and Chen, Y. (2023). Synthetic data generation for reinforcement learning: A survey. *arXiv preprint arXiv:2301.01234*.

Zhu, B., Jiao, J., and Jordan, M. I. (2024). Principled reinforcement learning with human feedback from pairwise or k-wise comparisons.

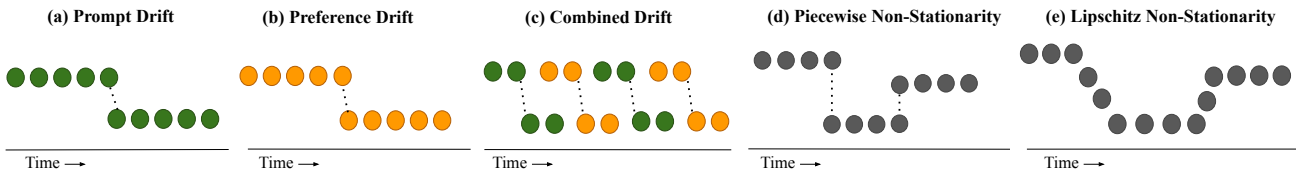


Figure 4: Non-stationarities in RLHF: (a-c) modes of drift and (d-e) types of drift. *(a)* Only the prompt distribution changes. *(b)* Only the preference distribution changes. *(c)* Both the prompt and the preference distributions change. *(d)* Piecewise non-stationarity. *(e)* Lipschitz non-stationarity.

Appendix

A. Lifelong RLHF

In section 2, 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*.

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 4 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.* □

A.1. Pseudocode

Algorithm 1 Lifelong RLHF

Initialize policy $\pi_\theta \leftarrow \pi^{\text{SFT}}$
for each time step $t = 1, 2, \ldots$ **do**
  Collect data batch $\mathcal{D}_t$ (e.g., via Algorithm 1)
  Train reward model $r_{\phi,t}$ on $\mathcal{D}_t$
  Update policy π_θ using an RL 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.

Algorithm 2 Synthetic Preference Generation for Task T_t

Input: LLM $\mathcal{M}$; budget N_t; templates τ_{prompt}, τ_{response}, and τ_{judge}
Initialize dataset $\mathcal{D}_t \leftarrow \emptyset$
for $i = 1$ to N_t **do**
  Generate prompt $x \sim \mathcal{M}(\cdot \mid \tau_{\text{prompt}})$
  Generate responses $y_1, y_2 \sim \mathcal{M}(\cdot \mid x, \tau_{\text{response}})$
  Generate preference label $y_c, y_r \sim \mathcal{M}(\cdot \mid x, y_1, y_2, \tau_{\text{judge}})$
  Append (x, y_c, y_r) 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.

B. Additional Experiments

B.1. 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 3 of the main paper. Tables 3, 4, and 5 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 errors during structured binding. These tables offer a granular view of the data diversity and quality underpinning our lifelong RLHF benchmarks.

Table 3: 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 4: 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

Table 5: 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

B.2. Human Evaluation Heatmap

Classification heatmap (see Figure 5) shows that LLMs align well with human judgments, with ambiguity concentrated in *both*/*neither* cases—especially for ELI5, which may benefit from stronger preference conditioning in generation.

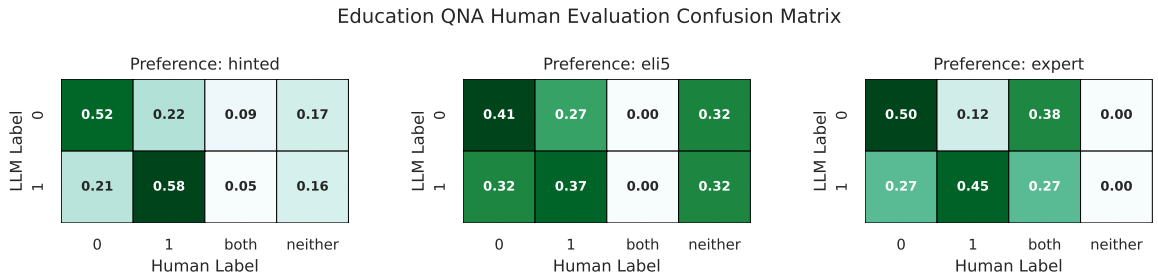


Figure 5: Confusion matrix comparing LLM (response 0 or 1) with human annotations (response 0, 1, *both*, *neither*). Most samples show agreement. When humans label *both* or *neither*, the LLM's predictions become more evenly split.

C. 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`)

D. 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 D.1, D.2, and D.3 display the prompts used respectively.

D.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.

D.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.

D.3. Data Generation Configurations

We show a representative sample of our configurations here. Our full set is available on HuggingFace and GitHub.

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`

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_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_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
```