浙 江 大 学

本 科 生 毕 业 论 文

文献综述和开题报告

学生姓名	张志心
学生学号	3210106357
指导教师	王灿
年级与专业	2021级计算机科学与技术
所在学院	计算机科学与技术学院

一、题目：基于对抗性偏好优化的语言模型对齐任务算法研究

二、指导教师对文献综述、开题报告、外文翻译的具体要求：

围绕面向数据高效的大语言模型对齐任务的对抗性偏好优化算法展开研究。首先，需调研与该课题相关的最新技术和研究成果，特别是对抗性训练、偏好优化、数据扩展和自我优化等方面的文献。要求阅读10～15篇中英文文献，选择最具代表性的一篇外文文献完成文献翻译。深入理解当前大语言模型对齐的研究现状，并在此基础上完成3000字以上的文献综述。文献综述需总结大语言模型对齐的基本思路与方法，结合个人理解，分析现有方法的优缺点。通过综述为研究目标和方法提供理论支持，并在此基础上撰写3500字以上的开题报告。报告应详细阐述该领域的基本理论、研究目标、方案及进度安排，展示对该领域的全面理解与研究规划。

项目的核心任务是设计一种创新的对抗性偏好优化算法，减少人工偏好数据的需求，提高大语言模型对齐任务的效率。研究过程中应通过文献学习和算法设计，培养严谨的研究作风，提升理论与实践结合的能力。最终，研究成果应为大语言模型对齐任务提供一种高效且数据需求较少的优化方案。

指导教师（签名）__________

年 月 日

目录

一、文献综述

1 背景介绍

大语言模型（Large Language Model/LLM），是一种人工智能模型，旨在理解和生成人类语言。它们在大量的文本数据上进行训练，可以执行广泛的任务，包括文本总结、翻译、情感分析等等。LLM 的特点是规模庞大，包含数十亿的参数，帮助它们学习语言数据中的复杂模式。这些模型通常基于深度学习架构，如 Transformer，这有助于它们在各种 NLP 任务上取得令人印象深刻的表现。

LLM 经过无监督训练后，可以得到知识和理解能力，但是很难去控制他们的生成行为。让模型做出符合人类偏好的回答是大模型对齐的重要任务[1-2]，要使这些模型更好地理解和符合人类的期望，确保它们的输出满足特定的偏好和需求，成为了一个关键问题。大多数传统的训练方法主要通过监督学习来优化模型的性能，但这种方式难以应对复杂的偏好对齐任务，尤其是在面对人类偏好时。为了解决这一问题，基于人类反馈的强化学习（RLHF）[3] 技术应运而生，它能够有效地将人类反馈融入到模型训练过程中，从而使得模型能够适应复杂的偏好要求。

RLHF 的核心思想是通过引入人类反馈，在训练过程中优化模型的策略，使其生成符合人类期望的输出。这个过程通常包括两个步骤：首先是通过收集人类反馈数据来构建奖励模型（Reward Model/RM），该模型用于评估不同输出的质量；然后，通过强化学习方法，使用 RM 来指导模型优化。传统上，RLHF 方法使用 PPO（近端策略优化）作为训练算法。PPO 通过引入奖励模型对生成输出进行评分，并通过策略梯度方法优化模型，使其逐步学习到最大化人类反馈奖励的策略[4]。这种方法在许多大型语言模型的训练中取得了显著的成功，推动了如 GPT-3.5、GPT-4 等模型的进一步发展[5-6]。RLHF/PPO 的具体步骤为：

1. 对预训练语言模型（LM）进行监督微调，得到 π^{SFT}。

2. 收集训练样本 (x, y_w, y_l)，其中 y_w 和 y_l 分别表示胜者和败者的响应。

3. 训练奖励模型 $r(x, y)$，用于对生成的文本进行评分：

$$\mathscr{L}_R(r_\phi, \mathscr{D}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathscr{D}} [\log \sigma(r_\phi(x, y_w) - r_\phi(x, y_l))] \tag{1-1}$$

4. 使用强化学习（RL）对策略模型 π_θ 进行微调，以优化奖励信号：

$$\max_{\pi_\theta} \mathbb{E}_{x \sim \mathscr{D}, y \sim \pi_\theta(y|x)} [r_\phi(x, y)] - \beta \mathbb{D}_{\mathrm{KL}} [\pi_\theta(y \mid x) \parallel \pi_{\mathrm{ref}}(y \mid x)] \tag{1-2}$$

5. 定期重新训练奖励模型 r 以适应新的数据分布。

尽管 PPO 在 RLHF 中取得了广泛应用，它也存在一些局限性，特别是在计算效率和扩展性方面。RLHF 的 pipeline 比较复杂，其中强化学习阶段需要多个模型，计算量巨大，耗内存。PPO 需要训练一个单独的奖励模型，这样不仅增加了计算成本，还可能引入模型与奖励模型之间的协同问题。为了应对这些挑战，近年来一些新的方法逐渐取代了 PPO，尤其是那些不依赖单独奖励模型的技术。这些方法主要依靠人类反馈的排名信息，减少了计算复杂度，并能够在更大规模上进行训练。例如，似然校准（SLiC）[7] 和通过排名对齐人类反馈（RRHF）[8]等方法，通过保留排名信息而非直接使用评分来进行模型优化，这样可以简化训练过程，并且提高了训练的效率。

离线对比学习和 DPO 等方法逐渐成为了主流。这些方法通过直接使用人类反馈数据进行训练，简化了训练过程并提高了效率。直接偏好优化（DPO）[9] 作为为目前最具前景的 RLHF 方法之一，它的最大特点是它直接在模型训练中使用人类的偏好数据，而无需依赖单独的奖励模型。与 PPO 方法不同，DPO 通过直接优化人类反馈数据来调整语言模型的策略，使其能够生成符合人类预期的输出。由于省去了奖励模型的构建和训练，DPO 方法不仅在计算上更为高效，而且避免了模型与奖励模型之间的协同问题，这使得它在实践中具有较大的优势。目前，DPO 已经被广泛应用于多个高性能模型的训练中，如 Zephyr、Mixtral 和 LLaMa-3 等模型[10-11]。

尽管 DPO 在模型训练中表现出色，但它的有效性仍然依赖于高质量的人类反馈数据。在一些版本的 DPO 中，仍然存在对奖励模型的间接依赖，尤其是在确定偏好数据时[12]。因此，如何在保证训练效率的同时，进一步提高 DPO 方法的鲁棒性，成为了未来研究的一个重要方向。同时，如何进一步优化这些方法，并解决分布匹配和 EBM 采样等问题，也仍然是未来研究的重要方向。

2 国内外研究现状

2.1 直接偏好优化算法 DPO

DPO 与 RLHF 拥有相同的训练目标。它可以直接从用户反馈信号（数据集）中进行学习，抛弃了传统的 RLHF 的技术流程：先学习 reward 模型、再用 RL 技术来进行优化，该算法被被证明拥有更好的稳定性和收敛性。该领域在近一年来吸引了大量世界一流的研究团队，并取得了飞速的发展。DPO 具有的优势明显，它可以快速根据用户偏好优化模型参数，计算消耗少。目前，DPO 已经被广泛应用于推荐系统，大模型优化领域。

在 RLHF 的训练公式上，DPO 首先识别到该其训练目标的显式最优解，

$$\pi_r(y \mid x) = \frac{1}{Z(x)}\pi_{\text{ref}}(y \mid x)\exp\left(\frac{1}{\beta}r(x,y)\right), \text{其中，} Z(x) = \sum_y \pi_{\text{ref}}(y \mid x)\exp\left(\frac{1}{\beta}r(x,y)\right) \tag{1-3}$$

尽管 RLHF 具有显式最优解，由于其中 $Z(x)$ 的计算成本很高，依旧不能直接使用。通过两侧取对数，化简后得到，

$$r(x,y) = \beta\log\frac{\pi_r(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta\log Z(x) \tag{1-4}$$

由此得到，可以通过对奖励函数使用模型参数进行重参数化，使得训练奖励模型与大语言策略模型的可以在同时进行，省去了强化学习的过程。因此，在 BT 模型下，DPO 的训练优化目标为，

$$\mathscr{L}_{\text{DPO}}(\pi_\theta;\pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathscr{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta\log\frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right)\right] \tag{1-5}$$

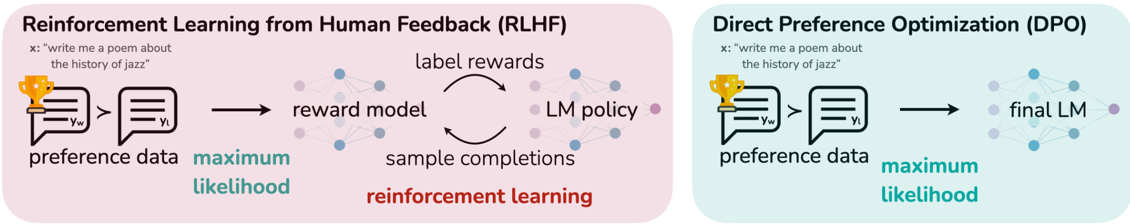


图 1.1 RLHF 和 DPO 的区别

通过将 RLHF 训练的 3，4 两步分别改用公式 1-4 和 1-5，即得到了更为轻便高效的 DPO 训练流程。图 1.1 展示了传统 RLHF 与 DPO 在训练上的过程对比。

2.2 利用 LLM 构建偏好数据

传统 RLHF/DPO 都需要大量人工的标注来训练奖励模型或直接在模型内进行偏好判断，为了实现高效且可扩展的对齐过程，近年来利用 LLM 进行偏好数据集构建受到关注。一种常见的方法是让 LLM 生成多个对同一输入提示的响应，并利用 LLM 的预测来近似人类对这些响应的偏好，这种技术通常被称为 *LLM-as-judge*[13-14]。然而，只有当 LLM 规模足够大且对齐良好时，该方法才能通过上下文学习有效地模拟人类偏好。另一种方法是使用外部奖励模型来替代人类偏好判断[15-16]，该方法可以更高效地进行偏好估计，但依赖于大量人类偏好数据来预训练奖励模型，并且在数据分布不匹配的情况下可能效果不佳。一些最新研究[16-19] 提出了结合迭代数据扩展与偏好学习的对齐方法，但它们通常依赖外部奖励模型或更强大的 LLM 进行偏好判断。相比之下，Dongyoung Kim 等发表的论文[20-22] 中提出了”Spread Preference Annotation”（SPA），它通过” 模型自判断 + 自修正” 的方式逐步扩展数据，从而在仅有少量人工偏好标注的情况下也能获得良好的模型对齐效果。论文的实验中仅利用训练中的 LLM 内在知识来进行新数据扩展和偏好学习。与常用的“外部奖励模型（RLAIF）”或用大模型做“LLM-as-judge”[13-14] 不同，论文强调让当前在训练的模型自身，根据它内部的输出概率（logits），直接判断新生成的响应哪个更好，即利用 DPO 的隐式奖励函数来对数据进行自标注。因此，SPA 不需要额外的强大外部模型来判断，也不必

依赖足够大型、已经高度对齐的人类偏好模型。此外，论文还提出了一种”自我修正”或”降噪”的策略，来处理模型在自己打标签时可能产生的噪声（即错误的偏好判断）。这一过程会识别出最有可能是错误标注的样本，并对它们的标注进行平滑或修正，从而减少被错误标签带偏的风险。

整个流程包括:

1. 基于模型现阶段的参数去生成新的回答对，再对这对回答进行模型自判断;
2. 然后将这些自标注的偏好数据与已有的小规模人工标注数据合并，进一步更新模型;
3. 如此反复，形成迭代的”扩充-学习”循环。

通过多次迭代，论文实验证明模型性能会显著提升。

2.3 存在问题

在前文已经讨论到，传统方法通常需要大量精心策划的偏好数据集。收集这样的数据集不仅代价昂贵且耗时，而且任何标签错误都可能在迭代对齐阶段传播[23]，导致模型行为的次优甚至不安全。这就提出了一个关键挑战：如何在保持对注释噪声的鲁棒性的同时，实现语言模型的偏好数据高效对齐?

从数据效率和鲁棒性的角度来看，现有的对齐方法通常面临两个主要问题:自我注释差距和噪声下缺乏平衡保证。SPA 中采用的自我注释[20-22]的方法（即让 LLM 为新的提示-响应对生成标签，而不是依赖人工标签）虽然确实降低了注释成本，但这些方法通常将策略更新和偏好注释视为不相关的过程。因此，一旦生成了带噪声的合成偏好，如果 LLM 的自标签嵌入了系统性的偏差或错误，这些错误可能会破坏未来的训练[24]，而没有有效的补救措施。一些对抗训练方法[25-26] 试图对抗偏好数据中的分布偏移，但它们通常缺乏正式的平衡保证，并且在实际中可能导致不稳定的优化循环。此外，这些对抗方法并未专门针对数据稀缺的对齐场景，因此在人工标签极其昂贵的情况下，其适用性有限。

3 研究展望

DPO 的训练范式具有很好的扩展性，而 SPA 的方法使得获取自动化的数据集成为可能，在此基础上，探索更好的训练算法，在减少对于人工标注偏好数据需求的同时，也能较好保持对注释噪声的鲁棒性，是一个重要的研究方向。

一种可能的思路是采用基于对抗式训练的偏好优化方法，以提高大语言模型对人类偏好的对齐效率，特别是在数据高效性方面。该思路需要考虑的问题主要有以下若干方面：首先，需要进一步优化对抗式训练过程中生成的对抗样本，结合模型的自标注和对抗性重加权来生成数据，进一步提升模型对真实世界应用中的异常情况和噪声数据的鲁棒性[27]；此外，未来还可以通过探索更高级的对抗训练算法，提升模型在处理多样化噪声时的稳定性，类似于 DPO 中处理潜在数据分布偏差的方式[9]；其次，对抗样本的支撑集也是需要考虑的一个因素，目前可能的思路是在已有数据的经验分布的临域内选取“最坏分布”（对于临域的选择，可以考虑限定半径为 ϵ 的 Wasserstein 球），以构造对模型最具有挑战性数据样本分布，在此基础上对模型进行不断的自增强训练，使得数据有效性和鲁棒性互相兼顾。此过程类似于主从博弈中的对抗优化方法[28]。此外，随着数据量的逐步增加，如何有效处理大规模的偏好数据并维持训练的高效性也是一个重要课题。当前自标注的生成过程在大规模数据集上可能会面临计算和存储上的挑战。因此，未来的研究可以考虑引入分布式训练和增量学习技术，使得对抗式训练能够在更大规模的数据集上进行有效的扩展[29]。

4 参考文献

[1] OUYANG L, WU J, JIANG X, et al. Training language models to follow instructions with human feedback[J]. Advances in Neural Information Processing Systems, 2022, 35: 27730-27744.
[2] BAI Y, JONES A, NDOUSSE K, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback[J]. ArXiv preprint arXiv:2204.05862, 2022.
[3] ZIEGLER D M, STIENNON N, WU J, et al. Fine-tuning language models from human preferences[J]. ArXiv preprint arXiv:1909.08593, 2019.
[4] ZIEGLER D M, STIENNON N, WU J, et al. Fine-tuning language models from human preferences[J]. ArXiv preprint arXiv:1909.08593, 2019.
[5] OUYANG L, WU J, JIANG X, et al. Training language models to follow instructions with human feedback[C]//Advances in Neural Information Processing Systems. 2022.
[6] YE J, CHEN X, XU N, et al. A Comprehensive Capability Analysis of GPT-3 and GPT-3.5 Series Models[Z]. 2023. arXiv: 2303.10420 [cs.CL].
[7] ZHAO Y, JOSHI R, LIU T, et al. Slic-hf: Sequence likelihood calibration with human feedback[J]. ArXiv preprint arXiv:2305.10425, 2023.
[8] YUAN Z, YUAN H, TAN C, et al. RRHF: Rank responses to align language models with human feedback without tears[J]. ArXiv preprint arXiv:2304.05302, 2023.
[9] RAFAILOV R, SHARMA A, MITCHELL E, et al. Direct preference optimization: Your language model is secretly a reward model[C]//Advances in Neural Information Processing Systems. 2023.
[10] TUNSTALL L, BEECHING E, LAMBERT N, et al. Zephyr: Direct distillation of lm alignment[J]. ArXiv preprint arXiv:2310.16944, 2023.
[11] JIANG A Q, SABLAYROLLES A, ROUX A, et al. Mixtral of Experts[J/OL]. CoRR, 2024, abs/2401.04088. arXiv: 2401.04088. https://doi.org/10.48550/arXiv.2401.04088. DOI: 10.48550/ARXIV.2401.04088.
[12] LIU T, ZHAO Y, JOSHI R, et al. Statistical rejection sampling improves preference optimization[J]. ArXiv preprint arXiv:2309.06657, 2023.
[13] BAI Y, JONES A, NDOUSSE K, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback[J]. ArXiv preprint arXiv:2204.05862, 2022.
[14] YUAN W, PANG R Y, CHO K, et al. Self-rewarding language models[J]. ArXiv preprint arXiv:2401.10020, 2024.
[15] JIANG D, REN X, LIN B Y. LLM-Blender: Ensembling Large Language Models with Pairwise Ranking and Generative Fusion[C]//Annual Conference of the Association for Computational Linguistics. 2023.
[16] Snorkel. New benchmark results demonstrate value of Snorkel AI approach to LLM alignment[J]. \group_end:https://snorkel.ai/new-benchmark-results-demonstrate-value-of-snorkel-ai-approach-to-llm-alignment, 2024.
[17] ROSSET C, CHENG C A, MITRA A, et al. Direct nash optimization: Teaching language models to self-improve with general preferences[J]. ArXiv preprint arXiv:2404.03715, 2024.
[18] WU Y, SUN Z, YUAN H, et al. Self-Play Preference Optimization for Language Model Alignment[J]. ArXiv preprint arXiv:2405.00675, 2024.
[19] XIONG W, DONG H, YE C, et al. Iterative preference learning from human feedback: Bridging theory and practice for rlhf under kl-constraint[C]//ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models. 2024.
[20] LEE H, PHATALE S, MANSOOR H, et al. RLAIF vs. RLHF: Scaling Reinforcement Learning from Human Feedback with AI Feedback[C]//Proceedings of the Forty-first International Conference on Machine Learning. 2024.
[21] YUAN W, PANG R Y, CHO K, et al. Self-Rewarding Language Models[C]//Proceedings of the Forty-first International Conference on Machine Learning. 2024.

[22] KIM D, LEE K, SHIN J, et al. Spread preference annotation: Direct preference judgment for efficient LLM alignment[C]//The Thirteenth International Conference on Learning Representations. 2025.

[23] CASPER S, DAVIES X, SHI C, et al. Open problems and fundamental limitations of reinforcement learning from human feedback[J]. ArXiv preprint arXiv:2307.15217, 2023.

[24] CHOWDHURY S R, KINI A, NATARAJAN N. Provably Robust DPO: Aligning Language Models with Noisy Feedback[C]//Proceedings of the Forty-first International Conference on Machine Learning. 2024.

[25] CHENG P, YANG Y, LI J, et al. Adversarial preference optimization[J]. ArXiv preprint arXiv:2311.08045, 2023.

[26] WU J, XIE Y, YANG Z, et al. Towards robust alignment of language models: Distributionally robustifying direct preference optimization[J]. ArXiv preprint arXiv:2407.07880, 2024.

[27] MOHAJERIN ESFAHANI P, KUHN D. Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations[J]. Mathematical Programming, 2018, 171(1): 115-166.

[28] BAŞAR T, OLSDER G J. Dynamic noncooperative game theory[M]. SIAM, 1998.

[29] VILLANI C, et al. Optimal transport: old and new[M]. Springer, 2009.

二、开题报告

1 问题提出的背景

1.1 背景介绍

1.1.1 大语言模型对齐的动机与挑战

随着深度学习和预训练范式的不断演进，大语言模型（LLM）已经成为人工智能领域中最引人注目的技术之一。诸如 GPT 系列、PaLM、LLaMA 等大模型在多种自然语言处理任务上取得了超越传统方法的成果。然而，伴随模型规模日益扩大，**对齐（alignment）**问题也愈发凸显。所谓“对齐”，即确保模型的行为与人类的价值观、道德观和期望保持一致，从而在减少错误或有害输出的同时，满足特定应用场景的需求。在对齐训练中最典型的思路，是通过**从人类反馈中强化学习**（RLHF, Reinforcement Learning from Human Feedback）来微调大模型，使其在回答问题时更贴近人类偏好。不过，这种方法依赖大量人工标注的偏好数据，不仅成本高昂，也难以在更大规模与更广泛领域上应用。与此同时，“人类偏好”本身也带有主观性，并非适用于所有任务场景；在某些有客观标准的领域，例如数学推理或编程任务，人工标注的效用并不一定最优。

与基于偏好的反馈信号相对应的，**基于规则反馈信号（Rule-based Feedback）**的方法作为另一种训练的范式，其依靠一套规则系统来为模型的输出进行评价，以减少对人类监督的依赖或，从而更高效、更广泛地对齐大模型的行为。所谓“规则反馈信号”，可由人工编写或模型自动生成的一系列准则、程序或约束，用来直接给模型的输出打分、施加惩罚或奖励。根据这些规则的形式与自动化程度，相关方法呈现出从**显式规则**到**隐式自监督**的丰富谱系。依赖客观标准的显式规则既能节省大笔人力标注，也能更精准地保证某些关键维度上的正确性；而偏向隐式方法的自监督方式，则在主观问题或大规模数据生成中具有强大的伸缩性与自动化优势。

大预言模型对齐人类偏好的训练面临着大量挑战。最核心的困难在于以下

几点。**大规模模型的复杂性：**现代大模型通常有数十亿到数千亿乃至万亿级参数，其内部表示和决策过程复杂且不透明，这种黑箱系统中很难施加并维持一组对齐约束，既要保证核心任务性能，又要避免产生与人类价值观相背的输出；**多样性与主观性：**大模型面临的任务和应用极度多样；有些问题存在客观唯一的正确答案（如编程、数学），另一些问题则充满主观性（如创意写作、社会价值），单一的人工偏好或简单的规则难以覆盖所有情形，导致对齐方法需要更具灵活性和适应性；**人力成本与可扩展性：** RLHF 等方法依赖数万甚至数十万条人工打分或标注数据，随着模型规模和应用领域扩张，纯粹依赖人工难以为继，迫切需要高效率、低人力消耗的对齐途径。

基于以上动机和挑战，研究者们开始尝试**“基于规则反馈信号”**的各种方法，希望借助显式、可编程的规则和隐式自监督的扩展技巧，实现更高效、更有针对性的对齐。

1.1.2 基于规则的反馈方法

基于规则反馈的主流方法包括**显式规则**、**规则 + 模型裁决**和 **隐式自监督信号**几种主要方法。

显式规则 指的是由人类手动编写或直接用程序实现的明确准则，用于判断模型输出的好坏、对错或合规/违规情况，常应用于数学推理（比对标准答案），编程生成（通过测试用例判断是否正确），事实问答（与标准知识库对照），以及安全过滤（关键词屏蔽、禁用政策等）。显式规则精确可控且无需额外设计奖励模型，但对于主观性较强的任务无法穷尽所有输出而难以胜任。

规则 + 模型裁决 往往在任务目标既包含客观要素，也涉及主观价值或复杂上下文时使用。在此类方法中，规则通常提供最低限度或关键维度的“硬性约束”，而模型自身（或额外训练的判别器）则在更主观、更模糊的维度上给出辅助评估，实现了一种“半自动”且相对可扩展的对齐流程。例如 AI 法官[1]、DeepMind Sparrow 等。半自动的规则有一定自动化程度，人工参与相对减少，但仍需人类设定或审校核心规则，且对规则理解偏差可能影响模型行为。

隐式自监督信号 则力图尽量脱离人工编写的具体规则或广泛的人类偏好数

据，转而让模型自行生成或推断对“更优”输出的偏好。例如 SPA[2]，LLM-as-Judge[3]，RLAIF[4] 等。这种模型大幅降低人力成本，可在无须大量人工标注的情况下迅速扩充偏好数据；但容易出现“自我偏见”、偏差放大、价值观漂移等风险，需要额外校准与审查。

在不同场景下，规则反馈方法呈现出一条从显式规则到隐式自监督的连续“频谱”。基于模型隐式自监督信号是从显式规则的良好延伸，包含了对合理性、正确性、安全性等既定规则等先验判断。

1.1.3 从归纳-演绎的角度看待基于规则的反馈方法

归纳是从具体的观测、实例或经验事实出发，通过总结、概括，得出更一般化或抽象化的规律、模型或假设。它通过观察若干具体例子总结共同特征或规律，从而提出较为通用的“规则”。然而，归纳得到的规律并不保证绝对正确，只是“相对可信”或“在目前观察的范围内大体成立”。今后若发现反例，需要修正或推翻。**演绎**则是从已有的公理、规则或理论出发，将其应用到具体场景或个案中去推导结论。它基于已有的普适法则推断出具体情况下的结论，若逻辑推理严谨且前提有效，所得结论必然正确。演绎依赖前提的正确性和完备性，若前提（公理或规则）本身有缺陷，则推理结论同样会受到影响。在新知识的产生过程中，归纳与演绎往往是相辅相成的：我们先通过归纳（从数据、现象中得出一个“可能的”规则），再在后续实践或应用（演绎）中验证、完善或修正该规则，这样便推动了对现有知识的扩展或对新知识的发现。

在“大语言模型对齐”或“机器学习模型训练”中，“基于规则的反馈”指的是预先由人类或程序所制定（或学得）的一组**规则**，用来对模型输出进行评估、打分、惩罚或奖励。此类规则有时是显式的（如编程测评的单元测试，安全过滤的关键词屏蔽），有时是半显式或隐式的（如由强模型、专家系统或复杂程序给出判定分值）。这些规则通常包含了对正确性、合规性或安全性的先验判断。无论是人工编写还是通过模型提炼，一开始往往会基于大量实例或经验做出总结——这就是**归纳**的过程。例如，在数学/编程评测中，人们归纳出若干“正确答案的判定标准”或“需要通过的测试用例”。在对话合规性中，安全准则也往往来

自过去对违规内容的收集、对社会规范的总结等。这些归纳出来的规则通常具有“一般性”或“抽象度”，使得它们能够适用到更广泛的情境中，而不仅仅局限于最初观测到的少数实例。一旦这些规则被定义或学得，系统（或模型）就能够利用这些规则对新输出进行评估，这就是**演绎**。从“通用规则”出发，针对具体问题或输出进行判断：若满足规则则给予奖励，不满足则惩罚。在运行时，对模型一次次的输出进行演绎式的“匹配或推理”：判定此刻的输出是否合规、正确或符合此前总结出来的规则。当反馈信号与模型的输出不匹配时，系统会在后续训练中进行**修正**（或人类会重新审视规则）。如果在演绎应用的过程中发现了足够多的反例，那么对原先规则的修订又会再次调用“归纳”环节的思维过程：根据新的实例或异常情况，完善或替换旧规则。在实际大模型的对齐或对抗训练中，模型的“错误”或“违规”输出起到“反例”的作用，让我们不断迭代规则或权重，从而使系统获得更高的精确度和泛化能力，这也是新知识（或更好的规则）产生和积累的过程。

1.2 本研究的意义和目的

随着大语言模型在各个领域的广泛应用，其对齐度和任务适应性成为了关键挑战。传统的对齐方法通常依赖大量人工偏好数据，这在许多实际应用中难以实现。现有方法往往面临数据稀缺和标注成本高的问题，尤其是当面对噪声数据时，模型的对齐效果可能会大打折扣。因此，如何在少量人工标注数据的基础上，提高模型对人类偏好的对齐效率，并确保对注释噪声具有足够的鲁棒性，成为了亟待解决的问题。

为了应对这一挑战，我希望提出一种基于对抗性训练的偏好优化算法，这是一种采用基于模型隐式规则的反馈进行自增强训练的方法，其中的对抗训练部分，结合了归纳-演绎法的思路，旨在构造具有挑战性的实例，激活模型不断的归纳新的知识，以适应人类偏好规则。本研究希望结合数据扩展与自我优化策略，旨在通过多轮对抗性训练，逐步优化大语言模型的对齐效果。通过生成对抗样本并进行自我标注，该方法能够减少人工偏好数据的需求，同时提升模型在面对噪声数据时的鲁棒性。此外，在对抗性训练的过程中，模型能够调整自身

的偏好判断能力，确保在任务适应性上表现出更高的稳定性与可靠性。

这一方法的最大意义在于，它不仅可以减少对大量人工标注数据的依赖，还能在噪声环境中有效保持模型的对齐精度，尤其在数据稀缺或标注困难的场景中，具有广泛的应用前景。通过引入对抗性训练机制，能够提升模型的自我纠正能力，增强其在实际任务中的适应性，确保偏好数据的高效利用与模型对齐的持续优化。通过这种方式，能够在保证模型鲁棒性的同时，实现更高效的数据利用和训练过程，从而为大语言模型的实际应用提供一种更加可行且经济的解决方案。

2 项目的主要内容和技术路线

2.1 主要研究内容

本研究主要围绕对抗性偏好优化算法的设计与实现展开，通过引入对抗性训练机制，结合数据扩展和自我优化策略，构建一个高效的偏好优化框架。核心思想是通过偏好反馈生成数据集，并在此基础上进行自标注和多轮迭代优化，从而提升模型的对齐效果。特别地，采用对抗性训练帮助模型在面对噪声数据时进行自我纠正，确保其在数据稀缺的情况下仍能保持高效的对齐。此外，希望形成一种结合数据扩展和自我优化策略的方法，通过现有的初始模型或参考模型生成更多的偏好反馈数据，并利用自我优化策略对这些数据进行多轮优化，以进一步提升模型的任务适应性和对齐度。

本研究的实验部分将通过一系列验证，展示该算法在数据稀缺和标注困难的应用场景中的有效性，尤其是在减少人工标注数据需求的同时，如何显著提升模型的对齐度和鲁棒性。

2.2 技术路线

2.2.1 对抗性偏好学习框架

本项目的大致算法框架如下:

1. 模型策略：大语言模型在训练时试图选择一条策略 π，以最大化在各种反

馈分布下的期望奖励。

2. 偏好分布：假设存在一个最坏情况或对抗性的偏好分布，会根据模型策略来挑选最具挑战性的输入或偏好标注。

3. 对抗过程：在给定策略下做对抗性反馈响应，模型通过在最不利的偏好条件下依然优化奖励，从而获得鲁棒性与普适性。

对于最坏分布，考虑构造关于偏好数据对奖励的经验分布，在其 Wasserstein 距离不超过 ε（给定超参数）下寻找最可能降低模型奖励期望的偏好分布。这里我们认为“最具有挑战性”的数据指的是目前模型最易混淆的偏好数据对。

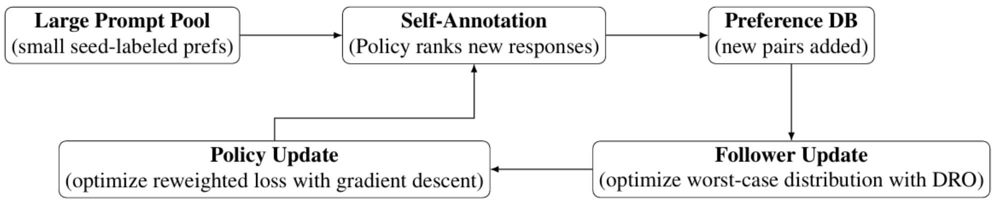


图 2.1 算法流程图

2.2.2 最坏分布求解方法

原始的 DPO 训练过程只考虑优化现有样本的经验分布，在样本数量较少的情况下容易出现过拟合的现象。在对抗性训练中，需要基于当前的数据经验分布构造一个使得模型期望奖励最低的“最坏分布”，对于这个任务，我考虑采用论文 DRO[5] 的技术路线，将求解过程看作一个高维随机规划问题。在温和假设下，Wasserstein 球上的分布随机优化问题实际上可以被重新表述为有限凸规划——在许多有趣的情况下，甚至可以作为可处理的线性规划。

设偏好数据分布为 $(x,y_w) \sim \mathscr{D}_w$，非偏好数据分布为 $(x,y_l) \sim \mathscr{D}_l$，其联合分布为 $\mathscr{D}$，样本集为 $D_n = \{(x,y_w,y_l)\}_1^N$。训练的优化的目标为：（其中 θ 是模型的参数）

$$J^* := \inf_{\theta\in\Theta} \left\{\mathbb{E}^{\mathscr{D}}[h(\theta,x,y_w,y_l)]\right\} \tag{2-1}$$

其中 $h(\theta, x, y_w, y_l) = -\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta\log\frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right)\right]$

我们考虑对问题进行适当的简化，在固定模型参数 θ 的情况下，定义

$$\xi = \sigma\left(\beta\log\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta\log\frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right) \tag{2-2}$$

$\xi \in \Xi$ 的边际分布可以由数据的经验分布 $\hat{\mathscr{D}}_N$ 导出，设为 $\hat{P}_N$，我们的目标是求解其 Wasserstein ε-临域 $\tilde{\mathscr{P}}_N := \mathbb{B}_\varepsilon(\hat{P}_N)$ 内最坏分布的目标期望：

$$\sup_{\mathbb{Q}\in\tilde{\mathscr{P}}_N} \mathbb{E}^{\mathbb{Q}}[\ell(\xi)] : \ell(\xi) = -\log\xi \tag{2-3}$$

假设不确定域 $\Xi \subseteq \mathbb{R}$ 是凸且闭的。且假设 $\ell(\xi) = \max_{k=1}^{K} \ell_k(\xi)$，其中 $\ell_k(\xi), k = 1, \cdots, K$ 是一系列凹函数，换言之，$\ell(\xi)$ 可以用一些列凹函数的 pointwise-maximun 表示（这样的假设并不会引入过多的限制）。最简单的，我们可以使用线性函数来构造 $\ell(\xi) = -\log\xi$ 的下凹包，可以证明该方法是对损失函数的一个良好近似。参考[5]的证明过程，我们可以证明构造最坏分布等价于求解如下有限凸优化问题：

$$\begin{aligned}
&\sup_{\alpha_{i,k}, q_{i,k}} \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}\alpha_{i,k}\ell_k(\xi_i - \frac{q_{i,k}}{\alpha_{i,k}}) \\
&\text{s.t.} \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}\|q_{i,k}\| \le \varepsilon \\
&\qquad \sum_{k=1}^{K}\alpha_{i,k} = 1 && \forall i \le N \\
&\qquad \alpha_{i,k} \ge 0 && \forall i \le N, \forall k \le K \\
&\qquad \xi_i - \frac{q_{i,k}}{\alpha_{i,k}} \in \Xi && \forall i \le N, \forall k \le K
\end{aligned} \tag{2-4}$$

对应最坏分布为：

$$\mathbb{Q}_r := \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}\alpha_{i,k}\delta_{\xi_i - \frac{q_{i,k}}{\alpha_{i,k}}} \tag{2-5}$$

其中 r 为迭代步数，在分布的轻尾假设条件下，可以证明 $r \to \infty, \mathbb{E}^{\mathbb{Q}_r}[\ell] \to \sup_{\mathbb{Q}\in\tilde{\mathscr{D}}_N}\mathbb{E}^{\mathbb{Q}}[\ell]$。这样我们就把最坏分布的构造这样一个高维随机规划转化为一个简单的凸优化问题。

2.2.3 在重加权数据上的训练

多轮训练过程可以参考 SPA[6] 论文中的算法框架，利用模型自生成自标注数据，经过 2.2.2 构造最坏分布，采用 DPO[7]的训练范式对重加权数据进行训练。

根据2.2.2，我们修改 DPO 的训练目标为如下形式：

$$\mathscr{L}_{\text{DPO}} = \sum_{i=1}^{N}\sum_{k=1}^{K} -\log\left[\alpha_{i,k}\left(\sigma\left(\beta\log\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta\log\frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right) - \frac{q_{i,k}}{\alpha_{i,k}}\right)\right] \tag{2-6}$$

2.3 可行性分析

首先，从理论可行性来看，迭代自增强模型已经在多个研究中证明了其有效性[2]，表明通过不断更新模型并利用其自身生成的偏好数据进行优化，能够显著提升模型的对齐性能。此外，分布鲁棒优化（DRO）[5] 提供了一种有效的框架来求解最坏情况下的偏好分布问题，其中高维随机规划问题可以等价转化为线性规划问题，降低了计算复杂度。针对凹凸性假设，可以使用线性函数拟合损失函数，从而保证最坏分布的可计算性。由于 DRO 已经在多个优化问题中得到成功应用，其方法在本研究中的适用性具有较强的理论支撑。

其次，从计算可行性来看，求解最坏分布通常是 DRO 方法的核心挑战，但已有研究表明，在适当的假设下，这一问题可以通过高效的线性规划求解[5]。特别是当数据量较大时，可以采用分组策略，以保证计算的可扩展性。结合高性能并行计算技术，优化过程可以在可接受的时间范围内完成，从而确保在大规模数据集上的训练效率。此外，现有的分布式计算框架（如 GPU 并行优化和参数服务器架构）可以进一步加速 DRO 求解，使得模型能够在实际应用场景中具备高效训练的能力。

最后，从实际应用可行性来看，当前基于 DPO 的方法已经展示了较好的偏好对齐能力，而本研究进一步结合自增强学习与对抗性优化，使得模型在减少人工标注需求的同时，也能提升对抗噪声数据的鲁棒性。这种方法适用于大规模真实世界场景，特别是人工标注数据昂贵且存在一定噪声的任务。在实际应用中，可以通过迭代自增强机制逐步优化偏好数据的质量，使得模型在较少的人类监督下依然能够达到高水平的对齐效果。

3 研究计划进度安排及预期目标

3.1 进度安排

1. 2025.2.28 以前，阅读有关大语言模型偏好学习相关论文，确定毕业论文选题。

2. 2025.3.15 以前，完成开题报告，文献综述和外文翻译，初步确定技术路线。

3. 2025.3.31 以前，复现参考文献 SPA 的算法流程，对于自增强的偏好学习范式进行更深入的理解。

4. 2025.4.15 以前，建模基于对抗性偏好学习的算法框架，并进行理论分析。

5. 2025.4.30 以前，完成算法设计和代码编写，并在现有数据集上跑通实验。

6. 2025.5.15 以前，完成更多补充实验，对算法进行更深入的评估。

7. 2025.5.25 以前，完成毕业论文的撰写，准备毕业答辩。

3.2 预期目标

1. 设计高效的偏好优化算法：通过引入对抗性训练机制，结合数据扩展和自我优化策略，优化大语言模型的对齐度，使其在少量人工标注数据的支持下，通过多轮迭代逐步提升实际任务中的表现。

2. 减少人工标注数据需求：利用自标注和偏好反馈生成机制，显著降低对大量人工标注偏好数据的依赖，使模型在数据稀缺的环境下仍能实现高效对齐，目标是在标注数据量减少至原始需求的 1/30 时，仍能取得较好性能。

3. 提高模型的鲁棒性和适应性：通过对抗性训练和自我优化策略，确保模型能够在面对噪声数据时进行自我纠正，并维持高适应性，尤其在不确定性较大的应用场景中保持较高的对齐效果。

4. 进行理论分析与性能评估：对所提出的算法进行系统的理论分析，确保其在面对数据噪声和偏差时具有鲁棒性，并进行大量实验验证其在不同任务和数据规模下的效果，确保算法具备广泛的实用性和推广价值。

5. 探索数据扩展与自我优化的潜力：进一步探索数据扩展和自我优化策略，提升模型在各种任务中的泛化能力和适应能力，从而增强其在不同应用场景中的表现和效率。

4 参考文献

[1] BAI Y, KADAVATH S, KUNDU S, et al. Constitutional ai: Harmlessness from ai feedback[J]. ArXiv preprint arXiv:2212.08073, 2022.

[2] KIM D, LEE K, SHIN J, et al. Spread preference annotation: Direct preference judgment for efficient LLM alignment[C]//The Thirteenth International Conference on Learning Representations. 2025.

[3] LI H, DONG Q, CHEN J, et al. LLMs-as-Judges: A Comprehensive Survey on LLM-based Evaluation Methods[J]. ArXiv preprint arXiv:2412.05579, 2024.

[4] LEE H, PHATALE S, MANSOOR H, et al. RLAIF vs. RLHF: Scaling Reinforcement Learning from Human Feedback with AI Feedback[C]//Proceedings of the Forty-first International Conference on Machine Learning. 2024.

[5] MOHAJERIN ESFAHANI P, KUHN D. Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations[J]. Mathematical Programming, 2018, 171(1): 115-166.

[6] KIM D, KIM J, LEE K, et al. Spread Preference Annotation: Direct Preference Judgment for Efficient LLM Alignment[C]//The Thirteenth International Conference on Learning Representations. 2025.

[7] RAFAILOV R, SHARMA A, MITCHELL E, et al. Direct preference optimization: Your language model is secretly a reward model[C]//Advances in Neural Information Processing Systems. 2023.

三、外文翻译: 扩散偏好标注：直接偏好判断高效对齐 LLM

论文标题：Spread Preference Annotation: Direct Preference JUDGMENT FOR EFFICIENT LLM ALIGNMENT

作者：Dongyoung Kim、Kimin Lee、Jinwoo Shin 等

摘要

对齐大型语言模型（LLMs）以符合人类偏好已成为实现最先进性能的关键步骤，然而，构建大规模的人类标注偏好数据集的成本极高。为了解决这一问题，我们提出了一种新的框架——扩散偏好标注 (Spread Preference Annotation, SPA)，该框架通过直接偏好判断来提升 LLM 的对齐能力，仅需极少量的人类标注偏好数据。我们的核心思想是利用小规模（种子）数据中的人类先验知识，并通过迭代生成响应和从自标注偏好数据中学习，不断提高 LLM 的对齐效果。具体而言，我们提出从 LLM 的 logits 中推导偏好标签，以显式提取模型的内在偏好。相比于使用外部奖励模型或隐式上下文学习的现有方法，我们观察到该方法明显更为有效。此外，我们引入了一种噪声感知偏好学习算法，以缓解生成偏好数据中潜在低质量信息的影响。实验结果表明，该框架能够显著提升 LLM 的对齐性能。例如，在 Ultrafeedback 数据中仅使用 3.3% 的真实偏好标签，我们在 AlpacaEval 2.0 评测中达到了比使用全部数据或最先进基线方法更优的对齐效果。

1　介绍

近年来，大型语言模型（LLMs）在各种自然语言处理（NLP）任务上取得了巨大进展，推动了代码助手和聊天机器人等被数百万用户使用的现实世界应用[1-3]。使 LLMs 与人类反馈对齐，尤其是通过学习人类偏好，被广泛认为是其成功的关键技术[4-6]。为增强这种对齐，各种偏好学习算法被广泛研究[7-8]。尽管取得了诸多进展，但当前仍面临的一个挑战是对大规模人类标注偏好数据的依赖。由于偏好数据的质量和数量对 LLMs 的成功对齐至关重要[9-10]，获取这些数据的巨大成本无疑构成了重大障碍。

为缓解这一挑战，让 LLMs 参与构建偏好数据并利用这些数据提升对齐能力的研究最近受到了关注。例如，一种典型的方法是针对输入提示生成多个响应，并通过 LLM 预测来近似人类偏好，这通常被称为 *LLM* 作为评判者（*LLM-as-judge*）[11-12]。然而，这些方法仅在 LLM 足够大且对齐良好时，才能有效地通过上下文学习模拟人类偏好。另一方面，使用外部奖励模型来高效替代人类偏

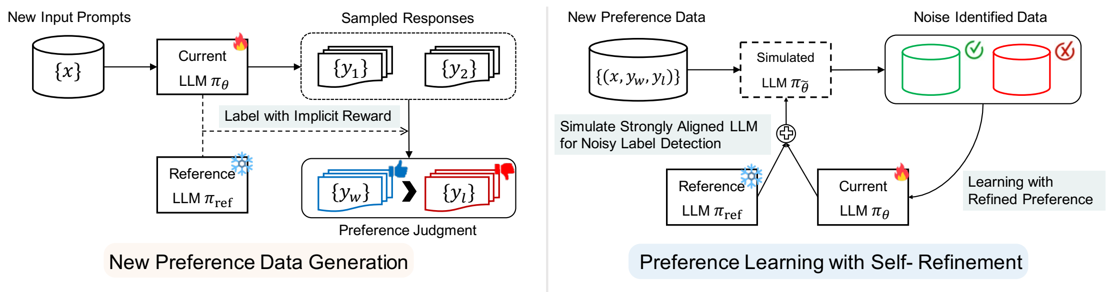


图 3.1 所提出的 SPA 框架的示意图。SPA 通过迭代 (1) 生成新的偏好数据和 (2) 在构造的数据上进行带有自我优化的偏好学习，逐步提高 LLM 的对齐程度。

好标注是一个可行方案[13-14]，但这种方法依赖于大量人类偏好数据，并且如果数据分布不匹配，也可能失效。此外，这些方法可能会受到 LLM 生成的潜在标注噪声的影响，而这一方面尚未得到充分研究。因此，在本研究中，我们的目标是提出一种方法，以克服上述限制，并在仅依赖少量人类标注的情况下，有效提高 LLM 的对齐能力。

贡献：我们提出了一个简单但有效的框架，称为 SPA，通过 **S**preading **P**reference **A**nnotation, SPA）的方式，仅依赖少量人类标注的偏好数据，利用**直接偏好判断**来提升 LLM 的对齐能力。我们的核心思想是在小规模（种子）数据的基础上，**逐步扩展**人类偏好知识，**通过迭代生成响应并利用自标注的偏好标签进行学习**，从而提升 LLM 的对齐效果。

具体而言，我们的技术贡献包括以下三个方面：

1. **直接从 LLM 的 logits 评判偏好标签**，以显式提取模型的内在偏好。这一方法比依赖外部奖励模型或隐式上下文学习的现有方法更为有效。

2. **引入基于置信度的偏好标签优化**，以降低生成数据中偏好学习的噪声风险。

3. **提出线性外推预测方法**，在当前模型与参考模型之间进行预测外推，以模拟更强对齐的模型预测，从而**提高噪声识别能力**，进一步增强偏好标签优化的效果。

我们通过使用少量人工标注的偏好数据对最新的 LLM 进行对齐，并在常用基准上评估其对齐程度，从而证明了所提出的 SPA 的有效性。例如，在 Ultrafeed-

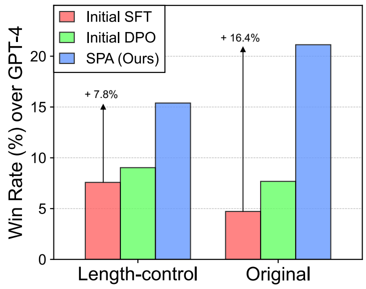


图 3.2 主要结果总结。在 AlpacaEval 2.0[15] 上的评估结果。我们的框架在没有额外人类偏好数据的情况下显著提升了 LLM 的对齐程度。

back 数据[10] 中仅使用 3.3% 的真实偏好数据，并基于 mistral-7b-0.1v SFT 模型[16] 进行实验，我们的框架在 AlpacaEval 2.0[15] 上的胜率相比初始 SFT 模型提升了 16.4% 以上（见图 3.2）。此外，AlpacaEval 2.0 的长度控制胜率从 7.58% 提高至 15.39%, MT-bench 分数[17] 从 6.38 提升至 6.94。与 LLM-as-judge[17] 等偏好判断方法相比，甚至与最近在 AlpacaEval 2.0 基准上表现最优的强奖励模型 PairRM[13] 相比，我们的方法在所有指标上均取得了更优的表现。更有趣的是，所提出的 SPA 即使在没有初始人类偏好数据的情况下，也能成功提升各种 LLM 的对齐程度。这些结果表明，我们的框架在实际应用中极具竞争力和实用性。

2 相关工作

LLM 与人类偏好的对齐。从人类偏好中学习已成为最新 LLM 的核心组成部分[1,3,18-19]，用于使其响应更符合用户的意图和价值观[6-7]。可以说，最流行的框架之一是基于人类偏好的强化学习（RLHF）[4-5]，其首先训练奖励模型，然后通过 KL 散度正则化微调 LLM 以最大化该奖励，从而防止 LLM 的奖励过度优化。另一方面，近年来提出了多种偏好学习算法，以更高效地利用人类偏好微调 LLM[8,20-25]。例如，Rafailov 等[8] 提出了直接偏好优化（DPO），其通过数学推导得出与 RLHF 等价的训练目标，从而无需单独训练奖励模型即可微调 LLM。Ethayarajh 等[20] 进一步消除了对成对偏好标签的依赖，基于人类效用模型构建训练目标。然而，这些方法都假设可用大量人工标注的偏好数据，而这通常需要巨大的数据获取成本。

利用 LLM 构建偏好数据。为了实现高效且可扩展的对齐过程，近年来利用 LLM 进行偏好数据集构建受到关注。一种常见的方法是让 LLM 生成多个对同一输入提示的响应，并利用 LLM 的预测来近似人类对这些响应的偏好，这种技术通常被称为 *LLM-as-judge*[9,12]。然而，只有当 LLM 规模足够大且对齐良好时，该方法才能通过上下文学习有效地模拟人类偏好。另一种方法是使用外部奖励模型来替代人类偏好判断[13-14]，该方法可以更高效地进行偏好估计，但依赖于大量人类偏好数据来预训练奖励模型，并且在数据分布不匹配的情况下可能效果不佳。一些最新研究[14,26-28] 提出了结合迭代数据扩展与偏好学习的对齐方法，但它们通常依赖外部奖励模型或更强大的 LLM 进行偏好判断。相比之下，我们仅利用训练中的 LLM 内在知识来进行新数据扩展和偏好学习。[29] 同时提出了一个类似的想法，重点关注长度正则化，而我们的工作则专注于通过所提出的自我优化过程提升生成数据的质量。

3 预备内容

设 LLM 为 π_θ，其针对给定的输入序列（*e.g.*, 提示词）x 生成输出序列（*e.g.*, 响应）y，即 $y \sim \pi_\theta(\cdot|x)$。我们的目标是使 π_θ 对各种输入提示提供符合人类偏好的响应。为此，我们采用广泛使用的偏好学习框架，该框架通过优化 π_θ 以学习人类对两个不同响应的偏好[4-5,7]。具体而言，我们假设偏好数据集 $\mathcal{D} = \{(x, y_l, y_w)\}$ 可用，其中包含输入提示 x、偏好响应 y_w 以及非偏好响应 y_l 的三元组。这里，偏好标签由真实标注者（通常是人类专家）进行标注。

奖励建模与强化学习微调。由于难以直接对 y_w 和 y_l 之间的成对偏好进行建模，常见的方法是引入奖励函数 $r(x, y)$，并基于 Bradley-Terry 模型[30] 对偏好进行建模：

$$p(y_w \succ y_l \mid x) = \frac{\exp\left(r(x, y_w)\right)}{\exp\left(r(x, y_w)\right) + \exp\left(r(x, y_l)\right)}. \tag{3-1}$$

根据该公式，可以通过最大似然目标函数对参数化的奖励模型 $r_\phi(x, y)$ 进行估计：

$$\mathcal{L}_R(r_\phi) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}}\left[\log \sigma\left(r_\phi(x, y_w) - r_\phi(x, y_l)\right)\right]. \tag{3-2}$$

其中，σ 为 sigmoid 函数。在完成奖励建模后，可以通过优化 LLM π_θ 以最大化 r_ϕ 所捕获的奖励，从而提升其对齐程度。通常，KL 散度正则项被引入，以防止 π_θ 过度优化奖励，该正则项相对于参考模型 π_{ref} 计算，并受超参数 $\beta > 0$ 控制[6-7]：[1]

$$\mathcal{L}_{\text{RLHF}}(\pi_\theta) = -\mathbb{E}_{y\sim\pi_\theta, x\sim\rho}\left[r_\phi(x,y)\right] + \beta \mathrm{D}_{\text{KL}}\left(\pi_\theta(y|x) \parallel \pi_{\text{ref}}(y|x)\right). \tag{3-3}$$

直接偏好建模与优化。 Rafailov 等[8] 提出了一种将 LLM π_θ 与偏好数据集 $\mathcal{D}$ 进行对齐的替代方法，称为直接偏好优化（DPO）。DPO 将奖励建模和强化学习微调这两个对齐步骤整合为一个统一的微调过程。具体而言，最优奖励函数可从 RLHF 目标（方程 3-3）推导出来，并基于目标 LLM π_θ 以及参考模型 π_{ref}[33-35]：

$$r(x,y) = \beta \log \frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z(x), \text{ where } Z(x) = \sum_y \pi_{\text{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r(x,y)\right). \tag{3-4}$$

随后，可以使用该奖励函数来衡量两个响应之间的偏好，并优化 π_θ 以最大化偏好数据集 $\mathcal{D}$ 中 y_w 相对于 y_l 的偏好：

$$p_\theta(y_w \succ y_l|x) = \sigma\left(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right). \tag{3-5}$$

$$\mathcal{L}_{\text{DPO}}(\pi_\theta) = \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[-\log p_\theta(y_w \succ y_l|x)\right]. \tag{3-6}$$

4 SPA：扩展偏好标注以提升 LLM 对齐性

概述。在本节中，我们介绍 SPA：**S**pread **P**reference **A**nnotation 通过直接偏好判断来对齐 LLM，同时降低构建偏好数据集的巨大成本。我们的核心思想是充分利用小规模（种子）数据中的人类先验知识，并逐步更新 LLM 以提升对齐性。具体来说，SPA 迭代执行两个步骤：（1）通过自生成偏好进行数据扩展（第 4.1 节）和（2）通过自优化偏好学习对 LLM 进行微调（第 4.2 节）。整体流程请参考图 3.1。

初始阶段：我们假设给定了一个小规模（种子）偏好数据集 D_0 和一个初始 LLM π_{init}。这里，我们遵循常见的做法[6-8]，使用已经在指令数据集上经过监督

[1] π_{ref} 通常由监督微调（SFT）LLM 初始化[31-32]，同时 π_θ 也初始化为 π_{ref}。

微调（SFT）的 LLM 作为 π_{init}[31-32]，但它尚未与人类偏好对齐。然后，我们首先通过在数据集 D_0 上使用 DPO[8]（方程 3-6）对 π_{init} 进行微调，从而获得弱对齐的大模型 π_0。在众多偏好学习方法中，我们选择 DPO，是因为它具有简单性和有效性。

4.1 通过自生成数据进行直接偏好判断来对齐 LLMs

对于第 i 次迭代（$i = 1, \ldots$），我们假设新的提示集 $X_i = \{x\}$ 已经可用，即 $X_i \cap X_j = \emptyset$ 对所有 $j = 0, \ldots, i-1$.[2] 从 X_i 中，我们通过使用 LLM 的内在生成和奖励建模能力构建第 i 次人工偏好数据集 $\mathscr{D}_i = \{(x, y_l, y_w) | x \in X_i\}$。具体来说，对于每个输入提示 $x \in X_i$，我们从 π_{i-1} 中采样两个响应 y_1 和 y_2，即 $y_1, y_2 \sim \pi_{i-1}(x)$，其中 π_{i-1} 是上一次迭代的结果模型。然后，利用通过 π_{i-1} 和 π_{init}（方程 3-4）捕获的奖励，我们测量 π_{i-1} 在 y_1 和 y_2 之间的偏好：

$$p_{i-1}(y_1 \succ y_2 | x) = \sigma\left(\beta \log \frac{\pi_{i-1}(y_1|x)}{\pi_{\text{init}}(y_1|x)} - \beta \log \frac{\pi_{i-1}(y_2|x)}{\pi_{\text{init}}(y_2|x)}\right). \tag{3-7}$$

然后，我们根据以下方式直接判断偏好标签，并通过此方式构建 $\mathscr{D}_i$：

$$(y_w, y_l) = (y_1, y_2) \text{ 如果 } p_{i-1}(y_1 \succ y_2 | x) > 0.5 \text{ 否则 } (y_w, y_l) = (y_2, y_1). \tag{3-8}$$

4.2 生成偏好数据的自优化以实现高效学习

在构建 $\mathscr{D}_i$ 之后，我们通过微调 π_θ 进行第 i 轮偏好学习，其中 π_θ 由 π_{i-1} 初始化，并使用 DPO 进行训练（此处，我们还将 π_{i-1} 作为方程 3-6 中的 π_{ref}）。学习自生成的偏好数据 $\mathscr{D}_i$ 可以利用 LLM 的能力有效地传播 $\mathscr{D}_0$ 中的人类偏好先验，从而提高对齐性。然而，这一过程也可能受到潜在标签噪声的影响，该噪声可能源于 X_i 的分布偏移或 π_{i-1} 在奖励建模方面的不足。因此，我们进一步提出了一种改进的偏好学习方法，利用一种新颖的去噪技术：通过**解耦噪声检测**进行偏好标签的**自优化**。

[2] $X_0 = \{x | (x, y_l, y_w) \in \mathscr{D}_0\}$

算法 3.1 SPA 算法

输入: 初始模型 π_{init}，种子偏好数据集 $\mathscr{D}_0$，迭代次数 T，新提示集 $\{X_i\}_{i=1}^{T}$，

使用 DPO 训练 π_{init} 和 $\mathscr{D}_0$（方程 3-6），获取初始弱对齐模型 π_0
for $t=1$ **到** T **do**
  通过 π_{t-1} 和 X_t 合成偏好数据 $\mathscr{D}_t$（方程 3-7 和 3-8）
  初始化训练模型和参考模型 $\pi_\theta \leftarrow \pi_{t-1}, \pi_{\text{ref}} \leftarrow \pi_{t-1}$
  for 小批量样本 $B \sim \mathscr{D}_t$ **do**
    $z_{\widetilde{\theta}} \leftarrow$ 对B 进行去耦噪声检测，基于$\pi_\theta, \pi_{\text{ref}}, X_t$（方程 3-11 和 3-12）
    计算训练损失 $\mathscr{L}_{\text{rf}}$，基于 $z_{\widetilde{\theta}}$ 和 π_θ 进行偏好标签优化（方程 3-10）
    更新模型参数: $\theta \leftarrow \theta - \eta \nabla_\theta \mathscr{L}_{\text{rf}}$
  end for
  初始化下一次迭代模型 π_t，使用更新后的参数 θ
end for
返回 π_T

偏好标签的自优化：我们的核心直觉是，推导出的偏好（方程 3-5）可以被视为当前训练中的 LLM π_θ 对 π_{i-1} 分配标签的置信度。当给定的响应对较难判别时，π_θ 可能表现出较低的置信度，表明存在较高的标签噪声概率。值得注意的是，置信度在噪声标签学习领域中是最常见的衡量指标之一[36-38]。基于这一直觉，我们首先识别置信度最低的 $K\%$ 样本：

$$z_\theta = 1 \ \text{如果}\ p_\theta(y_w \succ y_l|x) < \tau \ \text{否则}\ z_\theta = 0, \tag{3-9}$$

其中，τ 是数据集 $\mathscr{D}_i$ 中置信度排名第 K 百分位的样本的置信度。然后，利用该（潜在的）噪声识别标签 z_θ，我们采用标签平滑[39] 来优化偏好标签，以便在高噪声风险（即，$z_\theta = 1$）的情况下减少 π_θ 的训练置信度：

$$\mathscr{L}_{\text{rf}}(\pi_\theta) = \mathbb{E}_{(x,y_w,y_l)\sim\mathscr{D}_i}\left[-\left((1-\alpha * z_\theta)\log p_\theta(y_w \succ y_l|x) + \alpha * z_\theta \log p_\theta(y_l \succ y_w|x)\right)\right], \tag{3-10}$$

其中，α 是超参数。然后，我们使用 $\mathscr{L}_{\text{rf}}(\pi_\theta)$ 来训练 π_θ，以替代原始的 DPO 训练目标（方程 3-6）。

解耦噪声偏好检测：虽然使用优化后的偏好标签可以降低 π_θ 学习噪声偏好的风险，但由于用于噪声检测的模型 π_θ 来源于标签生成模型 π_{i-1}，其有效性可能受到限制。因此，为了进一步提升偏好标签优化框架的有效性，我们引入了解耦噪声检测（de-coupled noise detection）[36,40] 技术以对齐 LLM。具体而言，我们

通过模仿一个更强对齐的 LLM $\pi_{\widetilde{\theta}}$ 的偏好预测来识别偏好噪声：[3]

$$z_{\widetilde{\theta}}=1 \text{ 如果 } p_{\widetilde{\theta}}(y_w \succ y_l|x)<\tau \text{ 否则 } z_{\widetilde{\theta}}=0. \tag{3-11}$$

借助这种解耦标识方式，我们在方程 3-10 的基础上使用优化的偏好标签训练 π_θ，即，用 $z_{\widetilde{\theta}}$ 替换方程 3-10 中的 z_θ。这里，我们通过 π_θ 和 π_{ref} 的 logit 线性组合来近似计算 $\pi_{\widetilde{\theta}}$ 的 logit $h_{\widetilde{\theta}}$：[4] 这种方法受到最近研究的启发[41]，该研究表明，不同 β 值的 RLHF 对齐模型是参考模型和单一对齐模型的几何混合：

$$h_{\widetilde{\theta}}(x,y_{1:t-1})=(1+\lambda)*h_\theta(x,y_{1:t-1})-\lambda*h_{\text{ref}}(x,y_{1:t-1}), \tag{3-12}$$

其中，$\lambda>0$ 是超参数，$y_{1:t-1}$ 表示在第 t 个输出之前的输出序列。

值得注意的是，这种通过近似 $p_{\widetilde{\theta}}(y_w \succ y_l|x)$ 进行的解耦噪声检测不会增加额外计算开销，因为所需的 h_θ 和 h_{ref} 在计算原始 DPO 目标（方程 3-6）时已经获得。因此，SPA 仅需在原始 DPO 代码基础上添加少量额外代码即可实现。我们在算法 3.1 中展示了 SPA 的完整流程。

5 实验

在本节中，我们展示实验结果，以回答以下问题：

- SPA 是否仅利用少量人工标注的偏好数据即可提高 LLM 的对齐能力？（表 3.1，图 3.3）
- 我们提出的方法是否优于其他偏好标注方法？（表 3.2，图 3.4）
- SPA 是否在不同种类的种子数据和 LLM 设定下都具有泛化能力？（表 3.3，3.4，3.5）
- SPA 中的各个组件对最终效果的影响是什么？（表 3.6，3.7）

[3] 在方程 3-12 中的 λ 设定下，$\pi_{\widetilde{\theta}}$ 相当于通过方程 3-3 训练得到的模型，其 KL 项比 π_θ 小 $(1+\lambda)$ 倍。

[4] 当 $\pi_\theta(\cdot|x):=\text{Softmax}\big(h_\theta(x)\big)$ 时，我们将 $h_\theta(x)$ 视为 LLM π_θ 在给定输入 x 下的 logit。

5.1 实验设置

模型. 若未作特殊说明，我们的实验均使用经过监督微调的 Mistral-7b-0.1 模型[16] 作为初始模型 π_{init}（参见第 4 节）。具体而言，我们使用了开源模型[5]，该模型遵循 Zephyr[42] 训练策略，并在 Ultrachat[43] 指令数据上进行了微调。更多细节见附录。

基线方法. 为了评估所提出的偏好判断方法（方程 3-7）的有效性，我们与其他偏好判断方法进行了比较。具体而言，我们考虑了使用迭代 DPO（Iterative DPO）[14,23] 训练模型的基线方法，该方法通过 LLM 作为评判者（LLM-as-judge）[11,17]（即上下文学习）或外部强大奖励模型（如 PairRM[13]）来生成偏好数据并更新模型。值得注意的是，这些方法在去除 SPA 的自我优化（self-refinement）组件后，与 SPA 仅在评判方法上有所不同。详细内容见附录。

数据集. 在偏好学习任务中，我们使用 UltraFeedback 数据集[10]，与之前的研究保持一致[14,26]。[6] 具体而言，我们首先从该数据集中构建种子数据，包含 2K 个样本（占 60K 数据的 3.3%），每个样本包含提示（prompt）、模型生成的回复（response）和人工标注的真实偏好标签。在表 3.1 和 3.5 中，我们将 UltraFeedback 提供的真实偏好标签称为金标准标签（gold label）。然后，我们将剩余的数据划分为 8K、20K 和 30K 三个子集，仅保留提示部分。这些子集分别用于迭代阶段 1、2 和 3 的提示集。仅在表 3.3 的实验中，我们改变了种子数据的大小。

评估方法. 遵循 LLM 对齐的常见实践，我们主要采用以下两种评测方法来评估每个模型的表现：(1) AlpacaEval 2.0[15,44-45]。AlpacaEval 2.0 旨在近似评估模型在指令跟随任务中的人类偏好。该评测使用来自多个数据集的 805 条指令，并通过比较 GPT-4[18] 与测试模型的响应来计算胜率（win rate）。为了缓解 LLM 偏好的长度偏差问题[17,46]，我们同时测量原始胜率和长度控制（LC）胜率。其中，LC 胜率是通过一个单独训练的回归模型[45] 对响应长度影响进行中和后的调整胜率，以便更专注于质量评估。(2) 我们还使用 MT-Bench[17] 评估 LLM 的多方面能力。具体来说，MT-Bench 评测聊天机器人在多个类别（如数学、编程、角色

[5]alignment-handbook/zephyr-7b-sft-full

[6]"argilla/ultrafeedback-binarized-preferences-cleaned"

表 3.1 主要结果。在 AlpacaEval 2.0 和 MT-Bench 上，不同 Mistral-7B-v0.1 变体的评估结果。最佳分数以 加粗标出。

Models	Gold Label (%)	AlpacaEval 2.0 Len-control. Win Rate (%)	AlpacaEval 2.0 Win Rate vs. GPT-4 (%)	MT-Bench Avg. Score (0-10)
Mistral-7B-v0.1	-	0.17	0.50	3.25
Zephyr-7b-β	100	11.75	10.03	6.87
SFT	-	7.58	4.72	6.34
DPO	3.3	9.03	7.68	6.81
SPA (Ours)	3.3	**15.39**	**21.13**	**6.94**

扮演、写作等）上的综合能力。评测通过 GPT-4 对多轮对话问题的回答进行评分。这些基准测试可全面评估 LLM 在对齐人类偏好方面的表现及其实际应用中的整体有效性。

实现细节. 在初始化阶段之后，我们进行三轮数据扩展，每轮使用模型自生成的偏好数据。在数据扩展过程中，我们针对每个提示（prompt）独立采样 2 个响应，采样温度设为 0.7。然后，使用 SFT 模型作为参考模型，我们根据方程 3-7 进行偏好标注。为了获得初始模型 π_0，我们在种子数据集上进行了 3 轮 DPO 训练。在随后的每次迭代训练中，仅进行 1 轮训练。DPO 训练的超参数 β 设定为固定值 $\beta = 0.1$。批量大小设为 32，学习率为 5×10^{-7}。我们采用 AdamW 优化器，并使用余弦学习率调度器，其中 10% 的训练步骤作为预热阶段。对于 SPA 的超参数 α 和 $K\%$，我们使用固定值 $\alpha = 0.1$ 和 $K = 10$。此外，在去噪阶段，我们加入了预热机制，使去噪在完成 20% 训练步骤后才被激活。在去耦合噪声检测的超参数 λ 设定上，我们在迭代 1、2 和 3 轮中分别使用逐步递减的值 1/2、1/4 和 1/8。

5.2 主要结果

在完成 3 轮数据扩展和通过 SPA 进行微调后，训练得到的模型在 AlpacaEval 2.0 基准测试中的对 GPT-4 的胜率达到了 21.13%，如表 3.1 所示。相比于仅使用 3.3% 的标注数据并采用标准 DPO 训练时的 7.68% 胜率（7.68% $\rightarrow$ 21.13%），这

表 3.2 偏好判断基线比较。在 AlpacaEval 2.0 和 MT-Bench 上，使用不同偏好判断方法对迭代训练模型（从 SFT 模型开始）进行评估的结果。最佳分数以 加粗标出。

Methods	External Model	AlpacaEval 2.0 Len-control. Win Rate (%)	AlpacaEval 2.0 Win Rate vs. GPT-4 (%)	MT-Bench Avg. Score (0-10)
Iterative DPO (PairRM)	O	11.87	9.46	**6.98**
Iterative DPO (LLM-as-judge)	X	9.28	9.18	6.67
SPA (Ours)	X	**15.39**	**21.13**	6.94

一结果表明了显著的提升，同时长度控制胜率也有所提升（9.03% → 15.39%）。此外，SPA 在 MT-Bench 上获得了 6.94 的分数，明显优于使用相同 3.3% 标注数据进行 DPO 训练的模型（6.81）。更有趣的是，我们的方法在胜率（10.03% vs 21.13%）和长度控制胜率（11.75% vs 15.39%）方面均优于 Zephyr-7b-β，后者使用了相同的基础模型（Mistral-7B-0.1v）和 SFT 数据集，但采用了显著更多的标注偏好数据，即 UltraFeedback 数据集的 100%（相比之下 SPA 仅使用 3.3%）。在胜率上的这些显著提升清楚地表明了 SPA 所带来的整体性能增强。

接下来，在表 3.2 中，我们提供了额外的实验结果，以验证所提出的偏好判断方法。具体而言，表 3.2 中的三个实验可以被视为采用不同偏好判断方法的迭代 DPO 变体。从中可以观察到，SPA 相较于其他方法表现显著更优。具体来说，SPA 在 AlpacaEval 2.0 基准上对抗 GPT-4 的胜率达到 21.13%，相比之下，使用外部奖励模型 PairRM 作为基线的胜率仅为 9.46%。在长度控制胜率方面，SPA 也取得了 15.39% 的成绩，超越了奖励模型的 11.84%。

我们推测，使用训练中的 LLM 进行直接偏好判断的迭代 DPO 训练方法能够优于从外部奖励模型推断标签的情况，其主要原因与分布偏移（distribution shift）有关。随着迭代次数的增加，LLM 生成的数据分布会逐渐偏离种子偏好数据的分布。此时，外部奖励模型的有效性不可避免地下降，因为奖励模型是固定的，而生成的数据却越来越远离其训练分布。相比之下，SPA 通过不断更新的内在奖励模型生成偏好标签，因此在迭代过程中受到分布偏移的影响较小，从而在迭代训练中表现更为有效。

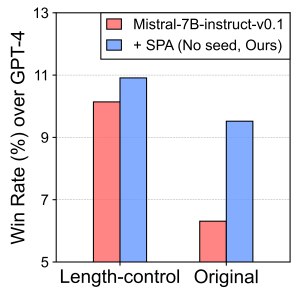


图 3.3 无种子数据的提升。在 AlpacaEval 2.0 上的评估结果，使用 Mistral-7B-instruct-v0.1 进行实验，并且 SPA 不使用任何种子偏好数据。

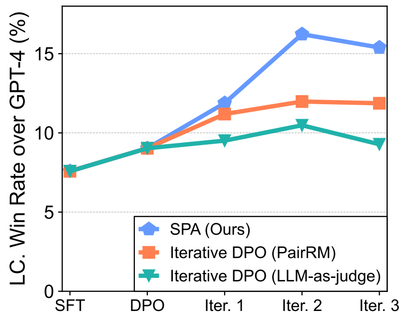


图 3.4 迭代过程中的提升。由 AlpacaEval 2.0 计算的长度控制（LC.）胜率（%）在 SPA 训练下持续提升，并优于其他基线方法。

关于这一点，我们在图 3.4 中的实验结果也提供了支持。在第 1 轮迭代中，两种方法的有效性相差不大。然而，在第 2 轮迭代时，性能差距显著拉大，这一结果从实验上支持了上述推论。

另一方面，使用上下文学习（in-context learning）方法的 LLM 作为判断器（LLM-as-judge）在胜率方面与 PairRM 相当，但在长度控制胜率方面表现较差（11.87% vs 9.28%），这表明 LLM-as-judge 方法存在局限性。总体而言，实验结果揭示了我们提出的直接偏好判断方法相较于其他判断方法的优越性。此外，这种优越性在多个迭代过程中均得到了一致验证，如图 3.4 所示。

5.3 更多分析

在本节中，我们通过在 AlpacaEval 2.0 上的实验结果对 SPA 进行额外分析。关于 MT-Bench 的更多比较以及附加实验的结果，请参见附录。

表 3.3 不同数量的种子数据。在 AlpacaEval 2.0 上，对使用不同数量的种子真实偏好标签训练的 Mistral-7B-v0.1 进行评估，比较 DPO 和 SPA 的表现。

	Used Ground-truth Preference Data			
Methods	0.8%	1.7%	3.3%	10%
DPO: LC Win Rate (%)	7.85	7.68	9.03	11.37
DPO: Win Rate (%)	5.53	5.49	7.68	9.32
SPA: LC Win Rate (%)	10.36	12.36	16.23	18.52
SPA: Win Rate (%)	11.34	13.72	19.94	23.79

不同数量种子数据的泛化能力在之前的实验中，我们假设初始仅有少量的人类偏好数据，例如 UltraFeedback 数据集的 3.3%。然而，SPA 的有效性并不依赖于种子偏好数据集的规模，我们通过额外的实验对此进行了验证。首先，我们通过改变种子真实偏好数据的比例来进行实验。具体而言，为了在每次迭代中使用固定的输入提示数据集，我们在实验中考虑了以下数据比例：[0.8%、1.7%、10%]。表 3.3 显示了在 Mistral-7B-v0.1 上进行 2 轮 SPA 训练后的 AlpacaEval 2.0 评测结果，其中也包括使用 3.3% 种子偏好数据的原始实验。从结果可以观察到，DPO 和 SPA 的对齐性能随着种子数据的增加而提升，且 SPA 始终优于 DPO，这表明 SPA 在不同种子偏好数据规模下均具有较强的鲁棒性。

此外，我们进一步评估了 SPA 在无种子偏好数据情况下的可行性。具体而言，我们希望回答一个问题：LLM 是否能够利用其在先前训练（如预训练或监督指令微调（SFT））过程中学习到的人类相关知识，在不同响应之间推导出明确的人类偏好。在该实验中，我们使用 Mistral-7b-instruct-0.1v[16] 作为初始模型（*i.e.*, π_0），并使用 Mistral-7b-0.1v-base 作为参考模型（*i.e.*, π_{init}）（关于初始设置，请参见第 4 节）。该设置允许我们验证，即使在没有种子偏好数据的情况下，当模型经过充分的迭代数据扩展和自我精调学习后，我们的方法仍然可以有效运行。如图 3.3 所示，胜率从 6.31% 提升至 9.79%，长度控制胜率从 10.14% 提升至 11.59%。这一结果表明，即使没有种子数据，SPA 仍然可以利用 LLM 内部信息来对齐人类偏好。

不同初始种子数据集的方差此外，我们进行实验来检验 SPA 对初始种子偏

表 3.4 不同的初始种子。在 AlpacaEval 2.0 上，针对不同初始种子偏好数据采样，对 Mistral-7B-v0.1 的不同变体进行评估。

Methods	1st Seed Data	2nd Seed Data	3rd Seed Data	Average	Variance
DPO: LC Win Rate (%)	9.03	8.74	9.54	9.10	0.16
DPO: Win Rate (%)	7.68	7.17	7.59	7.48	0.07
SPA (Ours): LC Win Rate (%)	16.23	13.77	16.38	15.46	2.10
SPA (Ours): Win Rate (%)	19.94	20.06	19.74	19.91	0.03

好数据集的敏感性，方法是使用不同的随机采样进行变化。在 SPA 训练 2 轮后的结果如表 3.4 所示。可以观察到，无论种子数据如何选择，SPA 始终能够提升对齐性能，并且不同种子数据之间的方差并不显著，特别是在标准胜率的情况下。尽管在长度控制（LC）胜率方面，我们的方法表现出相对较高的方差，但其最低置信区间值（13.36%）仍然明显高于最强基线模型（11.98%），这进一步验证了我们方法的有效性。

兼容不同模型接下来，为了验证我们框架在不同 LLM 上的兼容性，我们使用了三种不同的 LLM 进行实验：Phi-2-2.7B[47]、LLaMA3-8B[48] 和 Phi-3-14B。具体而言，我们基于这些模型的监督微调版本进行实验；对于 Phi-2，我们使用了在 UltraChat 数据集上微调的模型。[7] 对于 LLaMA-3[8] 和 Phi-3[9]，由于没有在 UltraChat 数据集上微调的版本，我们使用了它们的通用微调版本。这些实验的大多数设置与之前的实验保持一致，部分略有调整的实验设置详见附录 b.3。如表 3.5 所示，实验结果表明，将 SPA 应用于不同的 LLM，能够稳定提升其性能。例如，在 Phi-2 的实验中，相比 DPO，胜率从 5.67% 提升至 9.43%，长度控制胜率从 7.02% 提升至 9.1%。这些结果表明，SPA 的有效性不仅限于特定的 LLM，而是能够在不同的 LLM 上推广并获得稳定的性能提升。

消融研究为了评估自我优化（self-refinement）组件的影响，我们进行了消融实验，分别移除自我优化（SR）和解耦噪声检测（DND），并在此基础上对现有框架进行实验。实验结果如表 3.6 所示。当仅使用自我优化但不包含解耦噪声检

[7]lole25/phi-2-sft-ultrachat-full

[8]https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

[9]https://huggingface.co/microsoft/Phi-3-medium-4k-instruct

表 3.5 不同大模型的兼容性。在 AlpacaEval 2.0 上，使用不同训练方法（SFT、DPO 和 SPA）对各种大模型（Phi-2-2.7B、LLaMA-3-8B 和 Phi-3-14B）进行评估。最佳分数以 加粗标出。

Methods	Gold Label (%)	Phi-2-2.7B Len-control. Win Rate (%)	Phi-2-2.7B Win Rate vs. GPT-4 (%)	LLaMA-3-8B-Instruct Len-control. Win Rate (%)	LLaMA-3-8B-Instruct Win Rate vs. GPT-4 (%)	Phi-3-14B-Instruct Len-control. Win Rate (%)	Phi-3-14B-Instruct Win Rate vs. GPT-4 (%)
SFT	-	5.88	3.78	21.40	21.71	26.51	21.41
DPO	3.3	7.02	5.67	24.17	25.39	27.70	22.12
SPA (Ours)	3.3	**9.10**	**9.43**	**25.03**	**34.84**	**28.77**	**24.14**

表 3.6 消融实验。在 AlpacaEval 2.0 上，使用不同 SPA 方法配置下的迭代训练模型（从 SFT 开始）进行评估。DE、SR 和 DND 分别是数据扩展、自我优化和去耦噪声检测的缩写。最佳分数以 加粗标出。

Methods	DE	SR	DND	AlpacaEval 2.0 Len-control. Win Rate (%)	AlpacaEval 2.0 Win Rate vs. GPT-4 (%)
SFT	-	-	-	7.58	4.72
DPO	-	-	-	9.03	7.68
SPA (Ours)	O	X	X	14.41	19.91
	O	O	X	14.7	19.94
	O	O	O	**15.39**	**21.13**

测（参见公式 3-10）时，我们观察到性能略有提升，对 GPT-4 的胜率从 19.91% 轻微上升至 19.94%，长度控制胜率从 14.41% 提升至 14.7%。然而，当在自我优化过程中引入了解耦噪声检测（参见公式 3-11）后，性能得到显著提升，胜率从 19.91% 提升至 21.13%，长度控制胜率从 14.41% 提升至 15.39%。此外，这些结果表明，自我优化组件是提升性能的关键因素，有助于提高胜率并改善长度控制。

关于偏好判断方法的额外分析在表 3.7 中，我们进一步分析了偏好判断过程中参考模型的影响（参见公式 3-7）。该分析针对从第 1 轮到第 2 轮的训练过程进行，此时观测到了最显著的性能变化（即，我们从第 1 轮的模型开始微调）。为了单独比较不同判断方法的影响，我们遵循表 3.2 的设置，并排除了自我优化组件的影响。然后，我们设计了三种实验方案：(1) 使用上一轮的策略模型作为参考模型，(2) 不使用任何参考模型进行判断，(3) 使用 PairRM 进行判断。

表 3.7 附加分析。在 AlpacaEval 2.0 上，使用不同判断方法对从 SPA 第一次迭代结果模型进行微调的模型进行评估的结果。

Models	AlpacaEval 2.0 Len-control. Win Rate (%)	AlpacaEval 2.0 Win Rate vs. GPT-4 (%)
SPA after iteration 1	10.57	11.89
Eq. 3-7 with initial SFT model (Ours)	15.08	19.56
Eq. 3-7 with previous model	13.73	17.66
Judgment with PairRM	13.57	13.72
Judgment without reference model	12.83	12.35

实验结果如表 3.7 所示。结果表明，在 SPA 方法中，使用 SFT 模型作为参考模型进行偏好判断可以获得最高的性能提升。具体而言，使用上一轮的模型作为参考模型时，性能较低，且长度控制胜率下降幅度（15.08% vs 13.73%）大于胜率下降幅度（19.56% vs 17.66%）。尽管如此，该方法仍然优于使用 PairRM 进行判断。

这些结果可能表明，在训练过程中使用 LLM 本身进行偏好判断比使用外部模型更为重要，因为这样可以减少分布不匹配带来的影响。然而，在不使用参考模型的情况下（即，仅使用当前模型的似然值），其性能提升幅度是所有方法中最低的。这些结果进一步强调了选择合适的判断方法和参考模型对整体性能的显著影响。

6 结论

在本文中，我们提出了 SPA，一种可以通过最少的人类标注偏好数据高效提升大模型对齐度的方法。我们的主要贡献包括：开发了一种有效的数据扩展方法，结合了直接偏好判断方法，以及一种自我优化（潜在的）噪声偏好的偏好学习算法。我们通过在各种设置下微调近期的大模型，并在常用基准测试 AlpacaEval 2.0 和 MT-Bench 上进行评估，展示了 SPA 的有效性，并观察到显著的提升。我们期待 SPA 在未来的研究和实际应用中做出重要贡献，特别是在难以收集人类标注偏好的情况下。有关局限性和社会影响的讨论，请参见附录 A。

7 外文翻译参考文献

[1] Anthropic. Introducing the next generation of Claude[J]. \group_end:https://www.anthropic.com/news/claude-3-family, 2024.

[2] OpenAI. Introducing chatgpt[J]. \group_end:https://openai.com/blog/chatgpt, 2022.

[3] TEAM G, ANIL R, BORGEAUD S, et al. Gemini: a family of highly capable multimodal models[J]. ArXiv preprint arXiv:2312.11805, 2023.

[4] CHRISTIANO P F, LEIKE J, BROWN T, et al. Deep reinforcement learning from human preferences[C]//Advances in Neural Information Processing Systems. 2017.

[5] LEE K, SMITH L, ABBEEL P. Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training[C]//International Conference on Machine Learning. 2021.

[6] ZIEGLER D M, STIENNON N, WU J, et al. Fine-tuning language models from human preferences[J]. ArXiv preprint arXiv:1909.08593, 2019.

[7] OUYANG L, WU J, JIANG X, et al. Training language models to follow instructions with human feedback[C]//Advances in Neural Information Processing Systems. 2022.

[8] RAFAILOV R, SHARMA A, MITCHELL E, et al. Direct preference optimization: Your language model is secretly a reward model[C]//Advances in Neural Information Processing Systems. 2023.

[9] BAI Y, JONES A, NDOUSSE K, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback[J]. ArXiv preprint arXiv:2204.05862, 2022.

[10] CUI G, YUAN L, DING N, et al. Ultrafeedback: Boosting language models with high-quality feedback[J]. ArXiv preprint arXiv:2310.01377, 2023.

[11] BAI Y, KADAVATH S, KUNDU S, et al. Constitutional ai: Harmlessness from ai feedback[J]. ArXiv preprint arXiv:2212.08073, 2022.

[12] YUAN W, PANG R Y, CHO K, et al. Self-rewarding language models[J]. ArXiv preprint arXiv:2401.10020, 2024.

[13] JIANG D, REN X, LIN B Y. LLM-Blender: Ensembling Large Language Models with Pairwise Ranking and Generative Fusion[C]//Annual Conference of the Association for Computational Linguistics. 2023.

[14] Snorkel. New benchmark results demonstrate value of Snorkel AI approach to LLM alignment[J]. \group_end:https://snorkel.ai/new-benchmark-results-demonstrate-value-of-snorkel-ai-approach-to-llm-alignment, 2024.

[15] LI X, ZHANG T, DUBOIS Y, et al. AlpacaEval: An Automatic Evaluator of Instruction-following Models[Z]. https://github.com/tatsu-lab/alpaca_eval. 2023.

[16] JIANG A Q, SABLAYROLLES A, MENSCH A, et al. Mistral 7B[J]. ArXiv preprint arXiv:2310.06825, 2023.

[17] ZHENG L, CHIANG W L, SHENG Y, et al. Judging llm-as-a-judge with mt-bench and chatbot arena[C]//Advances in Neural Information Processing Systems. 2023.

[18] OpenAI. GPT-4 Technical Report[J]. ArXiv preprint arXiv:2303.08774, 2023.

[19] TOUVRON H, MARTIN L, STONE K, et al. Llama 2: Open foundation and fine-tuned chat models[J]. ArXiv preprint arXiv:2307.09288, 2023.

[20] ETHAYARAJH K, XU W, MUENNIGHOFF N, et al. Kto: Model alignment as prospect theoretic optimization[J]. ArXiv preprint arXiv:2402.01306, 2024.

[21] HONG J, LEE N, THORNE J. ORPO: Monolithic Preference Optimization without Reference Model[J]. ArXiv preprint arXiv:2403.07691, 2024.

[22] LIU T, ZHAO Y, JOSHI R, et al. Statistical rejection sampling improves preference optimization[J]. ArXiv preprint arXiv:2309.06657, 2023.

[23] XU J, LEE A, SUKHBAATAR S, et al. Some things are more cringe than others: Preference optimization with the pairwise cringe loss[J]. ArXiv preprint arXiv:2312.16682, 2023.

[24] ZHAO Y, JOSHI R, LIU T, et al. Slic-hf: Sequence likelihood calibration with human feedback[J]. ArXiv preprint arXiv:2305.10425, 2023.
[25] MENG Y, XIA M, CHEN D. Simpo: Simple preference optimization with a reference-free reward[C]//Advances in Neural Information Processing Systems. 2024.
[26] ROSSET C, CHENG C A, MITRA A, et al. Direct nash optimization: Teaching language models to self-improve with general preferences[J]. ArXiv preprint arXiv:2404.03715, 2024.
[27] WU Y, SUN Z, YUAN H, et al. Self-Play Preference Optimization for Language Model Alignment[J]. ArXiv preprint arXiv:2405.00675, 2024.
[28] XIONG W, DONG H, YE C, et al. Iterative preference learning from human feedback: Bridging theory and practice for rlhf under kl-constraint[C]//ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models. 2024.
[29] CHEN C, LIU Z, DU C, et al. Bootstrapping language models with dpo implicit rewards[C]//International Conference on Learning Representations. 2025.
[30] BRADLEY R A, TERRY M E. Rank analysis of incomplete block designs: I. The method of paired comparisons[J]. Biometrika, 1952, 39(3/4): 324-345.
[31] CHUNG H W, HOU L, LONGPRE S, et al. Scaling instruction-finetuned language models[J]. Journal of Machine Learning Research, 2024, 25(70): 1-53.
[32] WEI J, BOSMA M, ZHAO V Y, et al. Finetuned language models are zero-shot learners[C]//International Conference on Learning Representations. 2022.
[33] GO D, KORBAK T, KRUSZEWSKI G, et al. Aligning language models with preferences through f-divergence minimization[C]//International Conference on Machine Learning. 2023.
[34] PENG X B, KUMAR A, ZHANG G, et al. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning[J]. ArXiv preprint arXiv:1910.00177, 2019.
[35] PETERS J, SCHAAL S. Reinforcement learning by reward-weighted regression for operational space control[C]//International Conference on Machine Learning. 2007.
[36] HAN B, YAO Q, YU X, et al. Co-teaching: Robust training of deep neural networks with extremely noisy labels[C]//Advances in Neural Information Processing Systems. 2018.
[37] REED S, LEE H, ANGUELOV D, et al. Training deep neural networks on noisy labels with bootstrapping[J]. ArXiv preprint arXiv:1412.6596, 2014.
[38] SOHN K, BERTHELOT D, CARLINI N, et al. Fixmatch: Simplifying semi-supervised learning with consistency and confidence[C]//Advances in Neural Information Processing Systems. 2020.
[39] MÜLLER R, KORNBLITH S, HINTON G E. When does label smoothing help?[C]//Advances in Neural Information Processing Systems. 2019.
[40] LI J, SOCHER R, HOI S C. Dividemix: Learning with noisy labels as semi-supervised learning[C]//International Conference on Learning Representations. 2020.
[41] LIU T, GUO S, BIANCO L, et al. Decoding-time Realignment of Language Models[C]//International Conference on Machine Learning. 2024.
[42] TUNSTALL L, BEECHING E, LAMBERT N, et al. Zephyr: Direct distillation of lm alignment[J]. ArXiv preprint arXiv:2310.16944, 2023.
[43] DING N, CHEN Y, XU B, et al. Enhancing chat language models by scaling high-quality instructional conversations[J]. ArXiv preprint arXiv:2305.14233, 2023.
[44] DUBOIS Y, LI C X, TAORI R, et al. Alpacafarm: A simulation framework for methods that learn from human feedback[C]//Advances in Neural Information Processing Systems. 2023.
[45] DUBOIS Y, GALAMBOSI B, LIANG P, et al. Length-Controlled AlpacaEval: A Simple Way to Debias Automatic Evaluators[J]. ArXiv preprint arXiv:2404.04475, 2024.
[46] WANG Y, IVISON H, DASIGI P, et al. How far can camels go? exploring the state of instruction tuning on open resources[C]//Advances in Neural Information Processing Systems. 2023.
[47] LI Y, BUBECK S, ELDAN R, et al. Textbooks are all you need ii: phi-1.5 technical report[J]. ArXiv preprint arXiv:2309.05463, 2023.

[48] DUBEY A, JAUHRI A, PANDEY A, et al. The llama 3 herd of models[J]. ArXiv preprint arXiv:2407.21783, 2024.

四、外文原文

Published as a conference paper at ICLR 2025

SPREAD PREFERENCE ANNOTATION: DIRECT PREFERENCE JUDGMENT FOR EFFICIENT LLM ALIGNMENT

Dongyoung Kim[1], **Kimin Lee**[1], **Jinwoo Shin**[1], **Jaehyung Kim**[2]
[1]Korea Advanced Institute of Science and Technology , [2]Yonsei University
kingdy2002@kaist.ac.kr, jaehyungk@yonsei.ac.kr

ABSTRACT

Aligning large language models (LLMs) with human preferences becomes a key component to obtaining state-of-the-art performance, but it yields a huge cost to construct a large human-annotated preference dataset. To tackle this problem, we propose a new framework, **S**pread **P**reference **A**nnotation with direct preference judgment (SPA), that boosts the alignment of LLMs using only a very small amount of human-annotated preference data. Our key idea is leveraging the human prior knowledge within the small (seed) data and progressively improving the alignment of LLM, by iteratively generating the responses and learning from them with the self-annotated preference data. To be specific, we propose to derive the preference label from the logits of LLM to explicitly extract the model's inherent preference. Compared to the previous approaches using external reward models or implicit in-context learning, we observe that the proposed approach is significantly more effective. In addition, we introduce a noise-aware preference learning algorithm to mitigate the risk of low quality within generated preference data. Our experimental results demonstrate that the proposed framework significantly boosts the alignment of LLMs. For example, we achieve superior alignment performance on AlpacaEval 2.0 with only 3.3% of the ground-truth preference labels in the Ultrafeedback data compared to the cases using the entire data or state-of-the-art baselines.[1]

1 INTRODUCTION

Recently, large language models (LLMs) have made huge progress in various NLP tasks, leading to real-world applications that are used by millions of users, such as coding assistants and chatbots (Anthropic, 2024; OpenAI, 2022; Team et al., 2023). Aligning LLMs with human feedback, particularly through learning from human preferences, is widely considered a crucial technique for their success (Christiano et al., 2017; Lee et al., 2021; Ziegler et al., 2019). To enhance this alignment, various preference learning algorithms have been extensively explored (Ouyang et al., 2022; Rafailov et al., 2023). Despite these advancements, one of the remaining challenges is the reliance on large-scale human-annotated preference data. As the quality and quantity of the preference data are critical for the successful alignment of LLMs (Bai et al., 2022a; Cui et al., 2023), the huge cost to acquire such data inevitably presents significant obstacles.

To mitigate this challenge, engaging LLMs in constructing preference data and improving their alignment using these data has recently gained attention. For example, a representative way on this line is generating multiple responses for the input prompts, and then approximating human preference between them through LLM's predictions, often referred to as *LLM-as-judge* (Bai et al., 2022b; Yuan et al., 2024). However, these approaches are only effective when the given LLM is sufficiently large and well-aligned to mimic human preference via in-context learning. On the other hand, using an external reward model is considerable to substitute human preference annotation efficiently (Jiang et al., 2023b; Snorkel, 2024), but it is built on the availability of large human preference data and could also be ineffective if there is a distribution mismatch. Lastly, these approaches have a risk of potential labeling noise from LLMs, but this aspect has not been explored yet. Therefore, in this work, we aim to develop a method to effectively improve the alignment of LLM by overcoming these limitations but only relying on small human annotation.

[1]https://github.com/kingdy2002/SPA

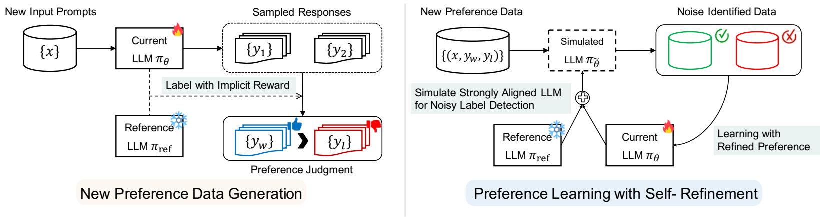


Figure 1: **Illustration of the proposed SPA framework.** SPA progressively improves the alignment of LLMs by iterating (1) the generation of new preference data and (2) the preference learning on the constructed data with self-refinement. Technical details are presented in Section 4.

Contribution. We introduce a simple yet effective framework, coined SPA, to improve the alignment of LLMs with only a small amount of human-labeled preference data, by **S**preading **P**reference **A**nnotation via direct preference judgment. Our key idea is to progressively expand the knowledge of human preference within the small (seed) data, by iteratively generating the responses and learning from them through the self-annotated preference labels. Specifically, our technical contributions are three-fold as described in what follows. First, we judge the preference labels directly using the logits of LLM to explicitly extract the model's inherent preference. This approach is more effective than previous methods that rely on external reward models or implicit in-context learning. Second, we introduce a confidence-based refinement of preference labels to reduce the risk of noise in preference learning with generated data. Third, to further enhance the effectiveness of this refinement, we propose using a linearly extrapolated prediction between current and reference models; it approximates predictions of a more strongly aligned model, leading to better noise identification.

We demonstrate the effectiveness of the proposed SPA by aligning recent LLMs with small human-annotated preference data and evaluating their alignment on the commonly used benchmarks. For example, using only 3.3% of ground-truth preference in Ultrafeedback data (Cui et al., 2023) with the mistral-7b-0.1v SFT model (Jiang et al., 2023a), our framework achieves over 16.4% increase in AlpacaEval2.0 (Li et al., 2023a) win rate compared to the initial SFT model (see Figure 2). Additionally, the AlpacaEval 2.0 length-controlled win rate is improved from 7.58% to 15.39%, and MT-bench score (Zheng et al., 2023) increased from 6.38 to 6.94. Compared to preference judgment methods like LLM-as-judge (Zheng et al., 2023), and even strong reward models such as PairRM (Jiang et al., 2023b), which have recently shown state-of-art performance in AlpacaEval2.0 benchmark, our approach consistently outperforms them across all metrics. More interestingly, the proposed SPA successfully improves the alignment of various LLMs, even without the initial human preference data. These results demonstrate that our framework is highly competitive and practical for real-world applications.

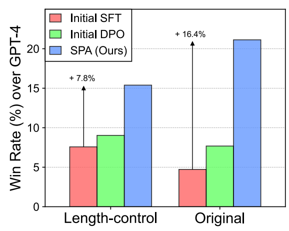


Figure 2: **Summary of main result.** Evaluation results on AlpacaEval 2.0 (Li et al., 2023a). Our framework significantly improves the alignment of LLMs, without additional human preference data. See detailed results in Section 5.

2 Related Work

Alignment of LLMs with human preference. Learning from human preferences now serves as a core component for the state-of-the-art LLMs (Anthropic, 2024; OpenAI, 2023; Team et al., 2023; Touvron et al., 2023) for aligning their responses with users' intent and values (Ouyang et al., 2022; Ziegler et al., 2019). Arguably, one of the most popular frameworks is reinforcement learning with human preference (RLHF) (Christiano et al., 2017; Lee et al., 2021), which first trains the reward model, and then fine-tunes LLM to maximize that reward with KL divergence regularization to prevent the reward over-optimization of LLM. On the other hand, various preference learning algorithms have recently been proposed to fine-tune LLMs with human preference more efficiently (Ethayarajh et al., 2024; Hong et al., 2024; Liu et al., 2023; Rafailov et al., 2023; Xu et al., 2023; Zhao et al., 2023; Meng et al., 2024). For example, Rafailov et al. (2023) proposes Direct Preference Optimization (DPO) which allows one to fine-tune LLMs without a separate reward modeling stage, by deriving the training objective mathematically equivalent to RLHF. Ethayarajh et al. (2024) further removes the reliance on pair-wise preference labels by formulating the objective based on a human utility model. However, these methods assume that large human-annotated preference data is available, which requires a huge data acquisition cost.

Engagement of LLMs for constructing preference data. For an efficient and scalable alignment procedure, engaging LLMs for preference dataset construction has recently received attention. One common approach involves generating multiple responses to input prompts from LLM, and using an LLM's predictions to approximate human preferences between them, a technique often referred to as *LLM-as-judge* (Bai et al., 2022a; Yuan et al., 2024). However, this method is effective only when the LLM is sufficiently large and well-aligned to mimic human preferences through in-context learning. Alternatively, employing an external reward model can efficiently replace human preference judgment (Jiang et al., 2023b; Snorkel, 2024), but this approach relies on the availability of extensive human preference data to pre-train reward model and may be ineffective if there is a distribution mismatch. Some concurrent works (Rosset et al., 2024; Snorkel, 2024; Wu et al., 2024; Xiong et al., 2024) have proposed the alignment procedure with iterative data expansion and preference learning. However, they use the external reward model or stronger LLM for the preference judgment. In contrast, we only utilize the intrinsic knowledge of training LLM for new data expansion and preference learning.

3 PRELIMINARIES

Let us denote LLM as π_θ, which generates an output sequence (*e.g.*, response) y for a given input sequence (*e.g.*, prompt) x, *i.e.*, $y \sim \pi_\theta(\cdot|x)$. Then, our goal is to make π_θ provide human-aligned responses to various input prompts. To this end, we consider the popular framework of preference learning, which optimizes π_θ to learn human preferences between two different responses (Christiano et al., 2017; Lee et al., 2021; Ouyang et al., 2022). Specifically, we assume that the preference dataset $\mathcal{D} = \{(x, y_l, y_w)\}$ is available which consists of the triplets of input prompt x, preferred response y_w, and dispreferred response y_l. Here, the preference labels were annotated by a ground truth annotator, that is usually a human expert.

Reward modeling and RL fine-tuning. Since a pairwise preference between y_w and y_l is hard to model directly, one of the common practices is introducing reward function $r(x, y)$ and modeling the preference based on this using the Bradley-Terry model (Bradley & Terry, 1952):

$$p(y_w \succ y_l \mid x) = \frac{\exp\left(r(x, y_w)\right)}{\exp\left(r(x, y_w)\right) + \exp\left(r(x, y_l)\right)}. \tag{1}$$

From this formulation, one can introduce a parametrized reward model $r_\phi(x, y)$ by estimating its parameters with the maximum-likelihood objective:

$$\mathcal{L}_R(r_\phi) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log \sigma\left(r_\phi(x, y_w) - r_\phi(x, y_l)\right)\right]. \tag{2}$$

where σ is a sigmoid function. After this reward modeling procedure, one could improve the alignment of LLM π_θ by optimizing it to maximize the reward captured by r_ϕ. Here, KL-distance from the reference model π_{ref} is usually incorporated as a regularization to prevent the reward over-optimization of π_θ, with a hyper-parameter $\beta > 0$ (Ouyang et al., 2022; Ziegler et al., 2019):[2]

$$\mathcal{L}_{\text{RLHF}}(\pi_\theta) = -\mathbb{E}_{y\sim\pi_\theta, x\sim\rho}\left[r_\phi(x, y)\right] + \beta \mathrm{D}_{\text{KL}}\left(\pi_\theta(y|x) \parallel \pi_{\text{ref}}(y|x)\right). \tag{3}$$

Direct preference modeling and optimization. Rafailov et al. (2023) propose an alternative approach to align LLM π_θ with the preference dataset $\mathcal{D}$, which is called Direct Preference Optimization (DPO). DPO integrates a two-step alignment procedure with reward modeling and RL fine-tuning into a single unified fine-tuning procedure. Specifically, the optimal reward function is derived from the

[2] π_{ref} is usually initialized with supervised fine-tuned (SFT) LLM (Chung et al., 2024; Wei et al., 2022a). Also, π_θ is initialized with π_{ref}.

RLHF objective (Eq. 3), with the target LLM π_θ and the reference model π_{ref} (Go et al., 2023; Peng et al., 2019; Peters & Schaal, 2007).

$$r(x, y) = \beta \log \frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z(x), \text{ where } Z(x) = \sum_y \pi_{\text{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r(x, y)\right). \quad (4)$$

Then, the preference between two responses could be measured using this reward derivation, and π_θ is optimized to maximize this preference of y_w over y_l using the preference dataset $\mathcal{D}$.

$$p_\theta(y_w \succ y_l|x) = \sigma\left(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right). \quad (5)$$

$$\mathcal{L}_{\text{DPO}}(\pi_\theta) = \mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}}\left[-\log p_\theta(y_w \succ y_l|x)\right]. \quad (6)$$

4 SPA: SPREAD PREFERENCE ANNOTATION TO BOOST ALIGNMENT OF LLMS

Overview. In this section, we present SPA: **S**pread **P**reference **A**nnoation via direct preference judgment to align LLMs while mitigating the huge cost for preference dataset construction. Our main idea is to fully exploit the human prior knowledge within the small (seed) data, and progressively update LLM to improve the alignment. To be specific, SPA iterates two steps: (1) data expansion with self-generated preference (Section 4.1) and (2) fine-tuning LLM with self-refined preference learning (Section 4.2). See Figure 1 for the overview.

Initial stage. We assume that a small (seed) preference dataset D_0 and an initial LLM π_{init} are given. Here, following the common practice (Ouyang et al., 2022; Rafailov et al., 2023; Ziegler et al., 2019), we use π_{init} which has been supervised fine-tuned (SFT) LLM on the instruction dataset (Chung et al., 2024; Wei et al., 2022a), but not aligned with human preference yet. Then, we first obtain weakly aligned LLM π_0 by fine-tuning π_{init} on D_0 using DPO (Rafailov et al., 2023) (Eq. 6). We adopt DPO among various preference learning methods due to its simplicity and effectiveness.

4.1 DIRECT PREFERENCE JUDGMENT TO ALIGN LLMS WITH SELF-GENERATED DATA

For the i-th iteration ($i = 1, \dots$), we assume that the new prompt set $X_i = \{x\}$ is available, *i.e.*, $X_i \cap X_j = \emptyset$ for all $j = 0, \dots, i-1$.[3] From X_i, we construct i-th artificial preference dataset $\mathcal{D}_i = \{(x, y_l, y_w)|x \in X_i\}$, by using LLM's intrinsic generation and reward modeling capabilities. Specifically, for each input prompt $x \in X_i$, we sample two responses y_1 and y_2 from π_{i-1}, *i.e.*, $y_1, y_2 \sim \pi_{i-1}(x)$ where π_{i-1} is the resulting model from the previous iteration. Then, using the reward captured with π_{i-1} and π_{init} (Eq. 4), we measure the preference of π_{i-1} between y_1 and y_2:

$$p_{i-1}(y_1 \succ y_2|x) = \sigma\left(\beta \log \frac{\pi_{i-1}(y_1|x)}{\pi_{\text{init}}(y_1|x)} - \beta \log \frac{\pi_{i-1}(y_2|x)}{\pi_{\text{init}}(y_2|x)}\right). \quad (7)$$

Then, we directly judge the preference label as below and construct $\mathcal{D}_i$ through this:

$$(y_w, y_l) = (y_1, y_2) \text{ if } p_{i-1}(y_1 \succ y_2|x) > 0.5 \text{ else } (y_w, y_l) = (y_2, y_1). \quad (8)$$

4.2 SELF-REFINEMENT OF GENERATED PREFERENCE DATA FOR EFFECTIVE LEARNING

After the construction of $\mathcal{D}_i$, we conduct i-th preference learning by fine-tuning π_θ, which is initialized by π_{i-1}, using DPO (here, we also use π_{i-1} as π_{ref} in Eq. 6). Learning the self-generated preference data $\mathcal{D}_i$ could improve the alignment by effectively spreading the human preference prior from $\mathcal{D}_0$ using the power of LLM. However, it also has a risk of the potential labeling noise which could occur from the distribution shift with X_i or insufficient reward modeling with π_{i-1}. Therefore, we further propose an improved preference learning method by introducing a novel denoising technique: *self-refinement* of preference labels with *de-coupled noise detection*.

[3] $X_0 = \{x|(x, y_l, y_w) \in \mathcal{D}_0\}$

Algorithm 1 SPA algorithm

Input: initial LLM π_{init}, seed preference dataset $\mathcal{D}_0$, number of improving iterations T, new prompt sets $\{X_i\}_{i=1}^{T}$,

Obtaining an initial weakly aligned model π_0 using DPO with π_{init} and $\mathcal{D}_0$ (Eq. 6)
for $t = 1$ **to** T **do**
  Synthesizing preference data $\mathcal{D}_t$ with π_{t-1} and X_t (Eq. 7 and 8)
  Initialization of training and reference models $\pi_\theta \leftarrow \pi_{t-1}, \pi_{\text{ref}} \leftarrow \pi_{t-1}$
  for mini-batch $B \sim \mathcal{D}_t$ **do**
    $z_{\widetilde{\theta}} \leftarrow$ De-coupled noise detection for B from $\pi_\theta, \pi_{\text{ref}}, X_t$ (Eq. 11 and 12)
    Calculate training loss $\mathcal{L}_{\text{rf}}$ with refined preference labels using $z_{\widetilde{\theta}}$ and π_θ (Eq. 10)
    Update model parameter: $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{rf}}$
  end for
  Initializing next iteration model π_t with the updated parameters θ
end for
return π_T

Self-refinement of preference label: Our key intuition is that one can view the derived preference (Eq. 5) can be viewed as the confidence of the currently training LLM π_θ for the labels assigned by π_{i-1}. Then, π_θ would exhibit lower confidence if the given pair of responses is uncertain to answer, indicating a higher probability of labeling noise. Notably, we also remark that confidence is one of the most popular metrics in the noisy label learning literature (Han et al., 2018; Reed et al., 2014; Sohn et al., 2020). Under this intuition, we first identify the $K\%$ least confident samples:

$$z_\theta = 1 \ \text{ if } \ p_\theta(y_w \succ y_l|x) < \tau \ \text{ else } \ z_\theta = 0, \tag{9}$$

where τ is the confidence of K percentile sample of $\mathcal{D}_i$. Then, with this (potentially) noise identification label z_θ, we refine the assigned preference label using label smoothing (Müller et al., 2019), to train π_θ less confidently when the risk of label noise is high (*i.e.*, $z_\theta = 1$):

$$\mathcal{L}_{\text{rf}}(\pi_\theta) = \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}_i}\left[-\big((1-\alpha * z_\theta)\log p_\theta(y_w \succ y_l|x) + \alpha * z_\theta \log p_\theta(y_l \succ y_w|x)\big)\right], \tag{10}$$

where α is a hyper-parameter. Then, we train π_θ using $\mathcal{L}_{\text{rf}}(\pi_\theta)$ instead of naive DPO (Eq. 6).

De-coupled noise preference detection: While learning with the refined preference label reduces the risk of learning π_θ the noisy preference, its effectiveness could be limited as the model π_θ for noise detection originated from the label generation model π_{i-1}. Therefore, to further improve the effectiveness of our preference label refinement framework, we introduce the de-coupled noise detection (Han et al., 2018; Li et al., 2020) technique for LLM alignment. Specifically, we identify the preference noise by mimicking the preference prediction of a more strongly aligned LLM $\pi_{\widetilde{\theta}}$: [4]

$$z_{\widetilde{\theta}} = 1 \ \text{ if } \ p_{\widetilde{\theta}}(y_w \succ y_l|x) < \tau \ \text{ else } \ z_{\widetilde{\theta}} = 0. \tag{11}$$

With this de-coupled identification, π_θ is trained with refined preference labels via Eq. 10 , *i.e.*, $z_{\widetilde{\theta}}$ is used to substitute z_θ in Eq. 10. Here, we obtain the prediction of $\pi_{\widetilde{\theta}}$ by approximating its logit $h_{\widetilde{\theta}}$ through the linear combination of the logits of π_θ and π_{ref}. [5] It is motivated by the recent work (Liu et al., 2024) that shows the aligned models via RLHF with varying β are geometric mixtures of a reference model and a single aligned model:

$$h_{\widetilde{\theta}}(x, y_{1:t-1}) = (1+\lambda) * h_\theta(x, y_{1:t-1}) - \lambda * h_{\text{ref}}(x, y_{1:t-1}), \tag{12}$$

where $\lambda > 0$ is a hyper-parameter and $y_{1:t-1}$ indicates the output sequence before t-th output.

We remark that this de-coupled noise identification by approximating $p_{\widetilde{\theta}}(y_w \succ y_l|x)$ *does not require additional computations* compared to DPO, since the required measurements h_θ and h_{ref} are obtained during the calculation of the original DPO objective (Eq. 6). Therefore, SPA only requires a few lines of additional code to the original DPO codebase. We present full procedure of SPA in Algorithm 1.

[4]With λ in Eq. 12, $\pi_{\widetilde{\theta}}$ is equivalent to model trained with $(1+\lambda)$ times smaller KL term than π_θ via Eq. 3.
[5]When $\pi_\theta(\cdot|x) := \text{Softmax}\big(h_\theta(x)\big)$, we refer $h_\theta(x)$ as the logit of LLM π_θ for the given input x.

5 Experiments

In this section, we present our experimental results to answer the following question:

- Does SPA improve the alignment of LLMs only using a small amount of human-labeled preference data? (Table 1, Figure 4)
- Does the proposed method outperform other preference labeling methods? (Table 2, Figure 3)
- Is SPA generalizable across various choices of seed data and types of LLMs? (Tables 3,4,5)
- What is the effect of each component in SPA? (Tables 6,7)

5.1 Experimental setups

Models. When there are no specific mentions, our experiments were conducted using the supervised fine-tuned Mistral-7b-0.1 model (Jiang et al., 2023a), as the initial model π_{init} in Section 4. Specifically, we use the open-sourced model[6] that follows the recipe of Zephyr (Tunstall et al., 2023) and fine-tuned on the instructions of Ultrachat (Ding et al., 2023). More details are in Appendix B.

Baselines. To evaluate the effectiveness of the proposed preference judgment method (Eq. 7), we compare it with other preference judgment methods. Specifically, we consider the baselines that train the model via Iterative DPO (Snorkel, 2024; Xu et al., 2023), which iteratively generate preference data and update the model, using LLM-as-judge (Bai et al., 2022b; Zheng et al., 2023) (*i.e.*, in-context learning) or an external powerful reward model (PairRM (Jiang et al., 2023b)) for the preference judgment. Notably, these approaches are the same in the case of changing the judgment method and removing self-refinement in SPA. Details are presented in Appendix B.

Datasets. For the preference learning dataset, we utilized UltraFeedback (Cui et al., 2023), following the previous works (Snorkel, 2024; Rosset et al., 2024).[7] To be specific, from this dataset, we first construct the seed data, consisting of 2K samples (3.3% of 60K) with prompts, responses, and ground truth preference labels. We refer the ground-truth preference label provided by the UltraFeedback as *gold label* in Tables 1 and 5. Then, the remaining samples are divided into subsets of 8K, 20K, and 30K samples, leaving only the prompts. These subsets were used as the prompt sets for the iteration stages 1, 2, and 3, respectively. Only for the experiments in Table 3, the size of seed data is changed.

Evaluations. Following the common practice in LLM alignment, we mainly evaluate each model our evaluations using (1) AlpacaEval 2.0 (Dubois et al., 2023; 2024; Li et al., 2023a). AlpacaEval 2.0 approximately evaluates human preference for instruction following. Using 805 instructions from various datasets, the evaluation is conducted by comparing the response of GPT-4 (OpenAI, 2023) and the testing model to measure win rates. To mitigate the length bias of LLM's preference (Wang et al., 2023b; Zheng et al., 2023), both original and length-controlled (LC) win rates are simultaneously measured. LC win rate is an adjusted win rate by neutralizing the effect of response length to focus on quality, using a separately trained regression model (Dubois et al., 2024). We also evaluate trained LLMs using (2) MT-Bench (Zheng et al., 2023) to assess different aspects of LLMs. Namely, MT-Bench evaluates a chatbot's overall abilities across multiple categories related to key LLM capabilities such as math, coding, roleplay, writing, etc. The evaluation is conducted by scoring responses to multi-turn questions using GPT-4. These benchmarks also provide a thorough evaluation of LLMs' alignment with human preferences and their overall effectiveness in practical applications.

Implementation details. After the initialization stage, we conduct three rounds of data expansion with self-generated preference data. For data expansion, we sampled 2 responses independently per each prompt with a temperature of 0.7. Then, using the SFT model as the reference model, we assign the preference label (Eq. 7). The initial DPO training to obtain π_0 was conducted for 3 epochs on the seed dataset. Training on each subsequent iteration was carried out for 1 epoch. For the hyper-parameter β of DPO, we used a fixed value of $\beta = 0.1$. The batch size was set to 32, and the learning rate was 5×10^{-7}. We employed AdamW optimizer and a cosine learning rate scheduler with a warm-up phase corresponding to 10% of the total training steps. For the hyper-parameters α and $K\%$ for SPA, we used fixed values of $\alpha = 0.1$ and $K = 10$. Additionally, a warm-up phase was included in the denoising stage, with denoising activated after 20% of the total training steps had been completed. Regarding the hyper-parameters λ for de-coupled noise detection, we utilized the progressively reduced values of 1/2, 1/4, and 1/8 for iterations 1, 2, and 3, respectively.

[6] `alignment-handbook/zephyr-7b-sft-full`

[7] `"argilla/ultrafeedback-binarized-preferences-cleaned"`

Table 1: **Main results.** Evaluation results on AlpacaEval 2.0 and MT-Bench with different variants of Mistral-7B-v0.1. The best scores are highlighted with **bold**.

Models	Gold Label (%)	AlpacaEval 2.0 Len-control. Win Rate (%)	AlpacaEval 2.0 Win Rate vs. GPT-4 (%)	MT-Bench Avg. Score (0-10)
Mistral-7B-v0.1	-	0.17	0.50	3.25
Zephyr-7b-β	100	11.75	10.03	6.87
SFT	-	7.58	4.72	6.34
DPO	3.3	9.03	7.68	6.81
SPA (Ours)	3.3	**15.39**	**21.13**	**6.94**

Table 2: **Comparison with baselines for preference judgment.** Evaluation results on AlpacaEval 2.0 and MT-Bench with iteratively trained models (from SFT model) under different preference judgment methods. The best scores are highlighted with **bold**.

Methods	External Model	AlpacaEval 2.0 Len-control. Win Rate (%)	AlpacaEval 2.0 Win Rate vs. GPT-4 (%)	MT-Bench Avg. Score (0-10)
Iterative DPO (PairRM)	✓	11.87	9.46	**6.98**
Iterative DPO (LLM-as-judge)	✗	9.28	9.18	6.67
SPA (Ours)	✗	**15.39**	**21.13**	6.94

5.2 Main results

After completing 3 iterations of data expansion and fine-tuning via SPA, the trained model achieved a 21.13% win rate against GPT-4 on the AlpacaEval 2.0 benchmark, as presented in Table 1. This represents a significant improvement compared to the 7.68% (7.68% $\rightarrow$ 21.13%) win rate achieved when using only 3.3% of labeled data with the standard DPO training, while the length-control win rate is also improved. (9.03% $\rightarrow$ 15.39%). In addition, SPA achieved a score of 6.94 on the MT-Bench, clearly outperforming the model trained with DPO (6.81) on the same amount of 3.3% gold labeling data. More interestingly, our framework achieved superior performance in both win rate (10.03% vs 21.13%) and length-control win rate (11.75% vs 15.39%), compared to Zephyr-7b-β which uses same base model (Mistral-7B-0.1v) and SFT dataset but uses significantly larger labeled preference data, *i.e.*, 100% of UltraFeedback dataset (v.s. 3.3% for SPA). These significant improvements in both win rates clearly affirm the overall enhancement in performance from SPA.

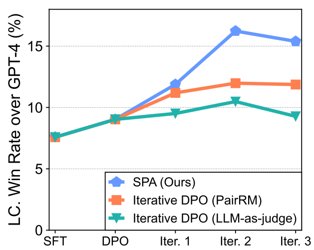


Figure 3: **Improvements during iterations.** Length control (LC.) win rate (%) measured by AlpacaEval 2.0 is consistently improved by SPA and it outperforms other baselines.

Next, in Table 2, we present additional experimental results to validate the proposed preference judgment method. Namely, three experiments in Table 2 can be viewed as the Iterative DPO variants with different preference judgment methods. One can observe that SPA showed significantly better performance compared to other methods. Specifically, SPA achieved a win rate of 21.13% against GPT-4 on AlpacaEval 2.0, compared to 9.46% for the baseline with an external reward model, PairRM. In terms of length control win rate, SPA achieved 15.39%, surpassing the reward model's 11.84%. Here, we conjecture that the reason why the Iterative DPO training with the proposed direct preference judgment method (using training LLM) outperforms the case with inferred labels from the external reward model is related to the distribution shift. As the iteration is increased, the distribution of the generated data with LLM is more shifted from the distribution of the seed preference data. Then, the effectiveness of the external reward model inevitably decreases, as the

Table 3: **Different number of seed data.** Evaluation results on AlpacaEval 2.0 with Mistral-7B-v0.1 trained with DPO and SPA under the different number of seed ground-truth preference labels.

	Used Ground-truth Preference Data			
Methods	0.8%	1.7%	3.3%	10%
DPO: LC Win Rate (%)	7.85	7.68	9.03	11.37
DPO: Win Rate (%)	5.53	5.49	7.68	9.32
SPA: LC Win Rate (%)	10.36	12.36	16.23	18.52
SPA: Win Rate (%)	11.34	13.72	19.94	23.79

reward model is fixed while the generated data is increasingly distant from its training distribution. In contrast, SPA generates the preference label using the intrinsic reward model that is continuously updated for each iteration. Therefore, it less suffers from the distribution shift during the iterations, and hence could be more effective for iterative training. Regarding this, we remark on the results in Figure 3; at iteration 1, the effectiveness of both approaches is not much different. However, the gap is significantly widened at iteration 2, and it empirically supports the above rationale.

On the other hand, the in-context learning approach (LLM-as-judge) shows a similar win rate compared to PairRM, but falls short in length control win rate (11.87% vs 9.28%), showing the limitations of the LLM-as-judge approach. Overall, the results reveal the superiority of our direct preference judgment over other judgment methods. Also, this superiority is consistently observed through the iterations, as shown in Figure 3.

5.3 More analyses

In this section, we conduct additional analyses of SPA by comparing the results on AlpacaEval 2.0. More comparisons on the MT-Bench and the additional experiments are presented in the Appendix.

Generalization across different numbers of seed data. Previously, we conducted the experiment by assuming that only a limited number of human preference data is initially given, *e.g.*, 3.3% of UltraFeedback dataset. However, the effectiveness of SPA does not depend on the size of the seed preference dataset and we validate this with the additional experiments. First, we conduct the experiments by varying the portion of the seed ground-truth preference data. Specifically, to use the fixed input prompt datasets for each iteration, we consider the following portions for the experiments: [0.8%, 1.7%, 10%]. Table 3 shows the results on AlpacaEval 2.0 with Mistral-7B-v0.1 after 2 iterations of training with SPA, including the original experiments with 3.3% seed preference data. Here, one can observe that the alignment performance under DPO and SPA is improved with the increased seed data, and SPA consistently outperforms DPO which demonstrates the robustness of SPA regarding the size of seed preference data.

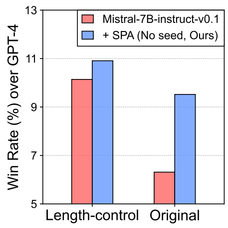


Figure 4: **Improvements without seed data.** Evaluation results on AlpacaEval 2.0 with Mistral-7B-instruct-v0.1 and SPA with no seed preference data.

We further evaluated the feasibility of using SPA even *without seed preference data*. Namely, we want to answer whether LLM can derive explicit human preference between responses, by leveraging their intrinsic knowledge learned about humans, during the previous training, such as pre-training or supervised instruction tuning (SFT). For this experiment, we used the Mistral-7b-instruct-0.1v (Jiang et al., 2023a) as the initial model (*i.e.*, π_0) and the Mistral-7b-0.1v-base as the reference model (*i.e.*, π_{init}) (see the initial setup in Section 4). This setup allows us to demonstrate that our framework can function effectively even in the absence of seed preference data, when the model is sufficiently fine-tuned with iterative data expansion and learning through self-refinement. As shown in Figure 4, the win rate increased from 6.31% to 9.79%, and the length-control win rate improved from 10.14% to 11.59%. This result indicates that SPA can leverage the internal information of LLMs to be aligned with human preference even without seed data.

Table 4: **Different initial seeds.** Evaluation results on AlpacaEval 2.0 with different variants of Mistral-7B-v0.1 under the different sampling of the initial seed preference data.

Methods	1st Seed Data	2nd Seed Data	3rd Seed Data	Average	Variance
DPO: LC Win Rate (%)	9.03	8.74	9.54	9.10	0.16
DPO: Win Rate (%)	7.68	7.17	7.59	7.48	0.07
SPA (Ours): LC Win Rate (%)	16.23	13.77	16.38	15.46	2.10
SPA (Ours): Win Rate (%)	19.94	20.06	19.74	19.91	0.03

Table 5: **Compatibility across various LLMs.** Evaluation results on AlpacaEval 2.0 with different training methods (SFT, DPO, and SPA) across various types of LLMs (Phi-2-2.7B, LLaMA-3-8B, and Phi-3-14B). The best scores are highlighted with **bold**.

Methods	Gold Label (%)	Phi-2-2.7B		LLaMA-3-8B-Instruct		Phi-3-14B-Instruct	
		Len-control. Win Rate (%)	Win Rate vs. GPT-4 (%)	Len-control. Win Rate (%)	Win Rate vs. GPT-4 (%)	Len-control. Win Rate (%)	Win Rate vs. GPT-4 (%)
SFT	-	5.88	3.78	21.40	21.71	26.51	21.41
DPO	3.3	7.02	5.67	24.17	25.39	27.70	22.12
SPA (Ours)	3.3	**9.10**	**9.43**	**25.03**	**34.84**	**28.77**	**24.14**

Variance with different initial seed dataset. In addition, we conduct experiments to check the sensitivity of SPA with the initial seed preference dataset by varying them with different random sampling. The results after 2 iterations of training with SPA are presented in Table 4. Here, one can observe that the proposed SPA consistently improves the alignment performance regardless of the given seed data, and the variance between them is not significant, especially in the case of a normal win rate. While ours exhibits a relatively high variance for length-controlled (LC) win rate, its lowest confidence interval value (13.36 %) is certainly higher than the value of the strongest baseline (11.98 %) which confirms the effectiveness of our method.

Compatibility with different models. Next, to verify the compatibility of our framework across various LLMs, we conducted experiments using three different LLMs: Phi-2-2.7B (Li et al., 2023b), LLaMA3-8B (Dubey et al., 2024), and Phi-3-14B. Specifically, we conducted experiments based on their supervised fine-tuned versions; for Phi-2, we used the model that has been fine-tuned on the UltraChat dataset like Mistral.[8] For LLaMA-3[9] and Phi-3[10], we used the generally fine-tuned models as there are no models that have been fine-tuned on the UltraChat dataset. Here, most of the experimental setups for these experiments are maintained, and the slightly adjusted setups are detailed in Appendix B.3. As shown in Table 5, the experimental results showed that applying SPA to various LLMs yields consistent improvements in the performance. For example, the win rate improved from 5.67% to 9.43%, and the length control win rate increased from 7.02% to 9.1%, in the case of Phi-2 after being trained with SPA compared to DPO. These results demonstrate that the effectiveness of SPA is not limited to the specific LLMs and is generalized across various LLMs.

Ablation study. To evaluate the impact of the self-refinement components, we conducted ablation experiments by excluding both self-refinement (SR) and decoupled noise detection (DND) from the existing framework. The results are presented in Table 6. With self-refinement without decoupled noise detection (Eq. 10), we observed a slight performance improvement, with the win rate against GPT-4 marginally increasing from 19.91% to 19.94%, and the length control win rate rising from 14.41% to 14.7%. But, when the decoupled noise detection is incorporated into the self-refinement (Eq. 11), we observed significant improvements, with the win rate increasing from 19.91% to 21.13% and the length control win rate improving from 14.41% to 15.39%. Also, these results confirm that the self-refinement component is a crucial factor in enhancing performance, contributing to both higher win rates and better length control.

Additional analysis with judgment methods. In Table 7, we further analyzed the impact of the reference model in the preference judgment process in Eq. 7. This analysis was conducted during

[8] lole25/phi-2-sft-ultrachat-full
[9] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct
[10] https://huggingface.co/microsoft/Phi-3-medium-4k-instruct

Table 6: **Ablation study.** Evaluation results on AlpacaEval 2.0 with iteratively trained models (from SFT) under different methodological configurations of SPA. DE, SR, DND are abbreviations of data expansion, self-refinement, and de-coupled noise detection, respectively. The best scores are highlighted with **bold**.

Methods	DE	SR	DND	AlpacaEval 2.0 Len-control. Win Rate (%)	AlpacaEval 2.0 Win Rate vs. GPT-4 (%)
SFT	-	-	-	7.58	4.72
DPO	-	-	-	9.03	7.68
SPA (Ours)	✓	✗	✗	14.41	19.91
	✓	✓	✗	14.7	19.94
	✓	✓	✓	**15.39**	**21.13**

Table 7: **Additional analyses.** Evaluation results on AlpacaEval 2.0 with models that fine-tuned with different judgment methods, from the resulting model of 1st iteration of SPA.

Models	AlpacaEval 2.0 Len-control. Win Rate (%)	AlpacaEval 2.0 Win Rate vs. GPT-4 (%)
SPA after iteration 1	10.57	11.89
Eq. 7 with initial SFT model (Ours)	15.08	19.56
Eq. 7 with previous model	13.73	17.66
Judgment with PairRM	13.57	13.72
Judgment without reference model	12.83	12.35

the transition from iteration 1 to iteration 2, where the most significant performance changes were observed (*i.e.*, we fine-tune from the resulting model of iteration 1). To isolate and compare the effect of judgment methods, we followed the setup in Table 2 and so excluded the influence of the self-refinement component. Then, we experimented with three setups by varying the judgment method using (1) the current policy from the previous iteration as the reference model, (2) performing judgment without any reference model, and (3) using the PairRM for judgment.

The results are presented in Table 7. Here, the experimental results demonstrated that the method used in SPA, where the SFT model was utilized as the reference model for preference judgment, achieved the highest performance increase. Specifically, using the model from the previous iteration as the reference model showed lower performance, with a relatively larger decrease in the length control win rate (15.08% vs 13.73%) compared to the win rate (19.56% vs 17.66%). Despite these decreases, it still outperforms using PairRM. These results may imply the importance of judging the preference through the training LLM rather than the external model, as it is less suffering from the distribution mismatch. However, without reference model (*i.e.*, only using the likelihood of the current model), the performance increase was the lowest compared to all other cases. These findings underscore the substantial impact of the choice of proper judgment method and reference model.

6 Conclusion

In this paper, we proposed SPA, a method that can efficiently improve the alignment of LLMs using minimal human-labeled preference data. Our main contributions include the development of an effective data expansion method with the direct preference judgment method and a preference learning algorithm with the self-refinement of (potentially) noise preference. We demonstrate the effectiveness of SPA by fine-tuning the recent LLMs with the various setups, and observing the significant improvements when evaluating them on the commonly used benchmarks, AlpacaEval 2.0 and MT-Bench. We expect SPA to make significant contributions to future research and practical applications, especially when the human-labeled preference is hard to collect. Limitations and societal impacts are further discussed in Appendix A.

Reproducibility Statement

For the reproducibility of our results, we have provided a detailed description of our methods and experimental setups in Section 5.1 and Appendix B. We also confirmed the robustness of our results through the experiment (Table 4). In addition, to further facilitate the reproduction, we will release our codes and the checkpoints for the trained models.

Acknowledgments

This work was mainly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2021-0-02068, Artificial Intelligence Innovation Hub, 50%; RS-2022-II220959, Few-shot Learning of Casual Inference in Vision and Language for Decision Making, 50%).

References

Anthropic. Introducing the next generation of claude. `https://www.anthropic.com/news/claude-3-family`, 2024.

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

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022b.

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*, 2017.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

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. *arXiv preprint arXiv:2310.01377*, 2023.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*, 2023.

Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. Rlhf workflow: From reward modeling to online rlhf. *arXiv preprint arXiv:2405.07863*, 2024.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

Dongyoung Go, Tomasz Korbak, Germán Kruszewski, Jos Rozen, Nahyeon Ryu, and Marc Dymetman. Aligning language models with preferences through f-divergence minimization. In *International Conference on Machine Learning*, 2023.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems*, 2018.

Jiwoo Hong, Noah Lee, and James Thorne. Orpo: Monolithic preference optimization without reference model. *arXiv preprint arXiv:2403.07691*, 2024.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023a.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. In *Annual Conference of the Association for Computational Linguistics*, 2023b.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Kimin Lee, Laura Smith, and Pieter Abbeel. Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In *International Conference on Machine Learning*, 2021.

Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations*, 2020.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval, 2023a.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*, 2023b.

Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. The unlocking spell on base llms: Rethinking alignment via in-context learning. In *International Conference on Learning Representations*, 2024.

Tianlin Liu, Shangmin Guo, Leonardo Bianco, Daniele Calandriello, Quentin Berthet, Felipe Llinares, Jessica Hoffmann, Lucas Dixon, Michal Valko, and Mathieu Blondel. Decoding-time realignment of language models. In *International Conference on Machine Learning*, 2024.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J Liu, and Jialu Liu. Statistical rejection sampling improves preference optimization. *arXiv preprint arXiv:2309.06657*, 2023.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. In *Advances in Neural Information Processing Systems*, 2024.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? In *Advances in Neural Information Processing Systems*, 2019.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. Webgpt: Browser-assisted question-answering with human feedback. In *arXiv*, 2021.

OpenAI. Introducing chatgpt. *https://openai.com/blog/chatgpt*, 2022.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

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

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Jan Peters and Stefan Schaal. Reinforcement learning by reward-weighted regression for operational space control. In *International Conference on Machine Learning*, 2007.

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

Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

Corby Rosset, Ching-An Cheng, Arindam Mitra, Michael Santacroce, Ahmed Awadallah, and Tengyang Xie. Direct nash optimization: Teaching language models to self-improve with general preferences. *arXiv preprint arXiv:2404.03715*, 2024.

Snorkel. New benchmark results demonstrate value of snorkel ai approach to llm alignment. `https://snorkel.ai/new-benchmark-results-demonstrate-value-of-snorkel-ai-approach-to-llm-alignment`, 2024.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*, 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. In *NeurIPS*, 2020.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

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

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*, 2023.

Peiyi Wang, Lei Li, Zhihong Shao, RX Xu, Damai Dai, Yifei Li, Deli Chen, Y Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *CoRR, abs/2312.08935*, 2023a.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, et al. How far can camels go? exploring the state of instruction tuning on open resources. In *Advances in Neural Information Processing Systems*, 2023b.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022a.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022b.

Yue Wu, Zhiqing Sun, Huizhuo Yuan, Kaixuan Ji, Yiming Yang, and Quanquan Gu. Self-play preference optimization for language model alignment. *arXiv preprint arXiv:2405.00675*, 2024.

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. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2024.

Jing Xu, Andrew Lee, Sainbayar Sukhbaatar, and Jason Weston. Some things are more cringe than others: Preference optimization with the pairwise cringe loss. *arXiv preprint arXiv:2312.16682*, 2023.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 2024.

Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J Liu. Slic-hf: Sequence likelihood calibration with human feedback. *arXiv preprint arXiv:2305.10425*, 2023.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, 2023.

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

A LIMITATION AND SOCIETAL IMPACT

A.1 LIMITATION AND FUTURE WORK

In the experiments, SPA has shown the tendency to increase the responses' length (please see Appendix D for the relevant results and discussions). We demonstrated that the improvement by SPA is not a simple result of such length increase, by observing the increase of win rate under a length length-controlled setup or MT-bench. However, depending on the user, this behavior could be dispreferred. In this sense, focusing on mitigating this bias during the self-improving alignment will be an interesting future direction, and can enhance the robustness and generalizability of SPA across more diverse scenarios.

A.2 SOCIETAL IMPACT

SPA enables efficient human preference learning, allowing for cost-effective training of models in data-scarce or domain-specific areas. Our framework supports alignment learning in various fields, including multilingual language learning and preferences beyond human helpfulness. Consequently, it could contribute to facilitating the widespread adoption of LLM technology across diverse sectors. By lowering the barriers to alignment learning, SPA makes it more accessible to a broader audience. However, the widespread availability of this technology also brings potential risks. The reduced cost of training models could enable malicious actors to misuse the technology, leading to societal issues. Therefore, it is crucial to implement ethical considerations and safety measures when deploying SPA technology to mitigate these risks.

B MORE DETAILS OF EXPERIMENTAL SETUPS

B.1 SFT MODEL SETUP

Mistral. For supervised fine-tuning, Ultrachat dataset (Ding et al., 2023) is used[11], batch size was set 128, total epoch was 1, and the learning rate was 2×10^{-5}. It employed Adam optimizer (Kingma & Ba, 2015) and a cosine learning rate scheduler with a warm-up phase corresponding to 10% of the total training steps.

Phi-2. For supervised fine-tuning, Ultrachat dataset is used, batch size was set 64, total epoch was 3, and the learning rate was 2×10^{-5}. It employed Adam optimizer and a cosine learning rate scheduler with a warm-up phase corresponding to 10% of the total training steps.

LLaMA-3 and Phi-3. As described in Section 5.3, we use the generically instruct-tuned versions for both LLaMA-3-8B and Phi-3-14B, as there are no SFT models tuned on Ultrachat dataset.

B.2 BASELINES EXPERIMENT SETUP

Zephyr-7b-β. We implemented Zephyr-7b-β (Tunstall et al., 2023), which is compared in Table 1, according to recipes. Our Zephyr-7b-β was trained using the same pre-trained model (mistral-7b-0.1v (Jiang et al., 2023a)) and the same SFT data (Ultrachat (Ding et al., 2023)), but there are marginal differences compared with recipes. We use SFT [12] models which trained with different recipes. Specifically, Zephyr-7b-β's SFT used the batch size of 512, but 128 was used for the ours SFT model. In addition, regarding the preference dataset, Zephyr-7b-β was trained using the original Ultrafeedback (Cui et al., 2023) [13] but we use cleaned version[14]. These changes in training data and the SFT model were aligned with SPA to ensure a fair comparison.

LLM-as-Judgement. For LLM-as-judge, we used an SFT model to employ Consitual AI's pairwise comparison prompt for judging preferences (Bai et al., 2022a). Preference is measured by comparing the logprob value of the token output as input to the following prompt (Listing 1). To ensure fair

[11] https://huggingface.co/datasets/HuggingFaceH4/ultrachat_200k
[12] https://huggingface.co/alignment-handbook/zephyr-7b-sft-full
[13] https://huggingface.co/datasets/HuggingFaceH4/ultrafeedback_binarized
[14] https://huggingface.co/datasets/argilla/ultrafeedback-binarized-preferences-cleaned

comparison and prevent low judgment performance, evaluation instructions were created using seed preference data which is the same form as Consitual AI's pairwise comparison. (Listing 2) Using these, additional SFT learning is performed to obtain an independent LLM-as-judge model. For this supervised fine-tuning, we set the batch size 32, total epoch is 3, and the learning rate was 2×10^{-5}. We employed Adam optimizer and a cosine learning rate scheduler with a warm-up phase corresponding to 10% of the total training steps.

Reward model judgment. For the reward model baseline, we selected PairRM (Jiang et al., 2023b) due to its high performance on AlpacaEval 2.0 (Snorkel, 2024; Wu et al., 2024). Unlike SPA, which was trained on only 2K gold label data, PairRM was trained on a large-scale dataset. The training data for PairRM includes the following:

- openai/summarize_from_feedback (Stiennon et al., 2020)
- openai/webgpt_comparisons (Nakano et al., 2021)
- Dahoas/synthetic-instruct-gptj-pairwise[15]
- Anthropic/hh-rlhf (Bai et al., 2022a)
- lmsys/chatbot_arena_conversations (Zheng et al., 2023)
- openbmb/UltraFeedback (Cui et al., 2023)

The total number of pairwise samples in this training data is approximately 500K, compared to 2K for SPA. Specifically, the summarize_from_feedback dataset contributes 179K samples, and the hh-rlhf dataset contributes 161K samples, making up a significant portion of the total.

B.3 Adjusted experimental setups for different LLMs

In Table 5, we conduct the experiments with different LLMs. As they exhibit different characteristics from the difference in backbone and sizes, we slightly adjusted the experimental setups while keeping most identical to the setups in Section 5.1.

Phi-2. We slightly adjust the learning rate to accommodate the different characteristics of the Phi-2 (5×10^{-6}). In addition, due to the smaller size of the Phi-2, we observe that performance improvements were not evident beyond iteration 2. Therefore, we present the results of iteration 1.

LLaMA-3 and Phi-3. We slightly adjust the learning rate to accommodate the different characteristics of models (1×10^{-5}). We conduct 1 epoch for the initial DPO training and maintain $\beta = 0.01$ throughout the entire training process. Since performance improvement has been only observed up to iteration 2 in Section 5.2. we conduct the experiments up to iteration 2 for these models.

B.4 Implementation details

Resources and computation cost. For all experiments, we utilized 4 A6000 GPUs. Under this computational resource, generating responses for 10K prompts takes approximately 1 to 2 hour, and preference judging for generated responses also takes about 1 to 2 hour. For training of model with Eq. 10, it takes about 1 to 2 hours per epoch. Therefore, the total time required to complete response generation, preference judgment, and one epoch of training was between 5 to 6 hours per 10K prompt.

Response generation. To mitigate the length bias from training with Direct Policy Optimization (DPO), we restricted the maximum token length for self-generated responses to 300 tokens. This limit corresponds to approximately 900 characters.

Refinement. Using the simple moving average (SMA) method, we measured the confidence over the most recent 50 samples. Based on this, we calculated the boundary corresponding to the lower K% of the confidence distribution (Eq. 11).

[15] https://huggingface.co/datasets/Dahoas/synthetic-instruct-gptj-pairwise

Table 8: **MT-Bench.** Evaluation results on MT-bench with different models. $\text{SPA}_{\texttt{inst}}$ and $\text{SPA}_{\texttt{phi}}$ denotes the fine-tuned Mistral-7B-Instruct-v0.1 and Phi-2 SFT with SPA, respectively.

Models	Gold Label (%)	Avg Score
Mistral-7B-Instruct-v0.1	-	6.78
$\text{SPA}_{\texttt{inst}}$ (Ours)	-	**7.12**
Phi-2 SFT	-	5.35
Phi-2 DPO	3.3	6.16
$\text{SPA}_{\texttt{phi}}$ (Ours)	3.3	**6.33**

```
f'''
Consider the following conversation between a human and an assistant:
[prompt]
Please choose the response that is the most helpful, honest, and
↪ harmless.
Options:
(A) [response 1]
(B) [response 2]
The answer is:
'''
```

Listing 1: Prompt used for judging preferences.

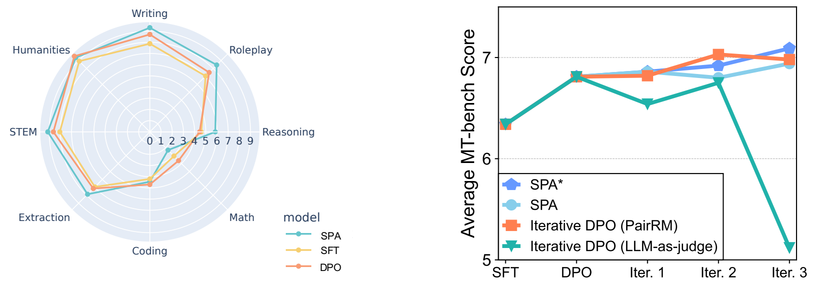
```
f'''
Consider the following conversation between a human and an assistant:
[Horses are hybrids of which two animals?]
Please choose the response that is the most helpful, honest, and
↪ harmless.
Options:
(A) [Horses, scientifically known as Equus caballus, are not the result
↪ of hybridization between two other animals. They are considered a
↪ species of their own within the Equidae family, which also includes
↪ other equines such as donkeys, mules, and zebras. Horses are
↪ believed to have evolved from a small, multi-toed forest-dwelling
↪ animal known as Eohippus, which lived around 60 million years ago.
↪ Over millions of years, these horses evolved into the large,
↪ single-toed animals we know today.]
(B) [Horses are not hybrids of any other animals. They are a unique
↪ species and are not a product of cross-breeding between different
↪ animals. Horses are domesticated animals and have been bred and
↪ developed over thousands of years by humans for various purposes
↪ such as transportation, riding, and agriculture.]
The answer is: (A)

'''
```

Listing 2: Evaluation instruction.

C More Evaluation Results in MT-Bench

In this section, we further provide the additional evaluation results on MT-Bench (Zheng et al., 2023), which couldn't be presented in the main text, due to the space issue.

We first present (a) task-wise evaluation results and (b) iteration-wise average improvement in Figure 5. As shown in Figure 5a, SPA consistently improves the performance in various tasks. Notably, there is almost no gain in Coding and degradation in Math. We remark that this phenomenon is



(a) MT-bench task-wise evaluation results

(b) Iteration-wise improvement with MT-bench

Figure 5: **MT-bench Evaluation.** More evaluation results with MT-bench.

Table 9: **Ablation study including MT-Bench.** Evaluation results on AlpacaEval 2.0 and MT-Bench with iteratively trained models (from SFT) under different methodological configurations of SPA. DE, SR, DND are abbreviations of data expansion, self-refinement, and de-coupled noise detection, respectively. The best scores are highlighted with **bold**.

Methods	DE	SR	DND	AlpacaEval 2.0 Len-control. Win Rate (%)	AlpacaEval 2.0 Win Rate vs. GPT-4 (%)	MT-Bench Avg. Score (0-10)
SFT	-	-	-	7.58	4.72	6.34
DPO	-	-	-	9.03	7.68	6.81
SPA (Ours)	✓	✗	✗	14.41	19.91	6.86
	✓	✓	✗	14.7	19.94	**7.09**
	✓	✓	✓	**15.39**	**21.13**	6.94

commonly observed in the relevant literature (Lin et al., 2024), which indicates that different training (Wang et al., 2023a) or inference (Wei et al., 2022b) schemes might be necessary to improve the performance in these tasks.

Next, in Figure 5b, one can observe that the average performance on the MT-bench is increased with more iterations. Specifically, while the Iterative DPO using PairRM shows the best performance until iteration 2, SPA* (without DND) outperforms it in iteration 3. It demonstrates the effectiveness of our framework for iteratively improving the alignment of LLM.

In addition, we measure the performances of Phi-2 variants and Mistral-7B-Instruct-v0.1 variants on MT-Bench in Table 8; these models are presented in Table 5 and Figure 4, respectively. As one can see, SPA consistently yields the improvement across different backbones of Mistral-7B-Instruct-v0.1 and Phi-2. Lastly, we present the full results of the ablation study (presented in Table 6) that includes the evaluation results on MT-Bench, in Table 9.

D More Quantitative Results

In this section, we present more quantitative results to demonstrate the effectiveness of SPA.

Mitigating length bias with SPA. Here, we provide a discussion of the relevant experimental results about the length bias present in SPA. During the experiments, we observe that LLMs trained with SPA tend to generate longer responses (see 10), which could be dispreferred depending on the user. Regarding this, we first emphasize that the improvement with SPA is not merely due to longer outputs, as shown by the significant gains in the length-controlled win rate in all experiments in Section 5.

Nevertheless, to further address the concerns regarding this issue, we further investigate whether previously researched length control techniques can be easily integrated into SPA. Specifically, we

Table 10: **SPA with length regularization.** Evaluation results on AlpacaEval2.0 with different variants of Mistral-7B-v0.1 from SPA and the additional length regularization term.

Models	Gold Label (%)	Len-control. Win Rate	Win Rate vs. GPT-4	Avg. len (# chars)
Mistral-7B-v0.1	-	0.17	0.50	5692
SFT	-	7.58	4.72	901
DPO	3.3	9.03	7.68	1802
Zephyr-7b-β	100	11.75	10.03	1552
SPA (Original, Iter. 1)	3.3	11.88	12.95	2150
SPA (Modified, Iter. 1)	3.3	11.39	12.31	**2013**
SPA (Original, Iter. 2)	3.3	16.23	19.94	2749
SPA (Modified, Iter. 2)	3.3	14.46	18.23	**2448**

Table 11: **LLM-as-Judgment with model from previous iteration.** Evaluation results on AlpacaEval 2.0 with different variants of Mistral-7B-v0.1.

Methods	Len-control. Win Rate (%)	Win Rate vs. GPT-4 (%)
LLM-as-judge (Iter. 1)	8.88	8.01
LLM-as-judge (Iter. 2, orig)	9.49	8.46
LLM-as-judge (Iter. 2, prev. init)	9.74	10.09
SPA (Iter. 2, ours)	**15.46**	**19.91**

apply the length penalty approach from RLHFlow Dong et al. (2024). This method heuristically reduces the reward model's reward based on the output length (Eq. 13) during preference labeling. We utilze hyperparamter α between 0.001 to 0.01 that minimize the length increase. The results, shown in Table 10, indicate that this modification successfully reduces the average length while largely preserving the performance improvements from SPA. These results demonstrate that SPA can be easily integrated with existing research related to length control.

$$r_{\texttt{penalty}}(x, y) = r(x, y) - \alpha|y| \tag{13}$$

LLM-as-Judge with the model from previous iteration. For the LLM-as-Judgement baseline, we used a fixed model fine-tuned specifically for evaluating preferences between responses, as described in Appendix B.2. On the other hand, at each iteration, the proposed SPA utilizes the previous iteration's model, π_{i-1}, for the direct preference judgment. Therefore, one can be curious how this difference affect to the performance.

To answer this, we conducted new experiments; at the 2nd iteration, the evaluation model was initialized with the resulting model from the 1st iteration and fine-tuned with the constructed preference evaluation dataset using the seed preference data (see details in Appendix B.2). The evaluation results (denoted as *LLM-as-Judge (Iter. 2, prev. init)*) on AlpacaEval 2.0 are presented in Table 11, along with other methods at the 2nd iteration. While this approach yielded improved alignment compared to the fixed model, SPA still significantly outperformed this baseline. This underscores that the superior performance of SPA arises from its novel preference evaluation techniques rather than the specific evaluation model used.

E Qualitative Examples

In this section, we present examples of the responses from the models trained under different methods (SFT, DPO, and SPA), for a qualitative comparison. The results are presented in Figures 6, 7, and 8. As one can see in the examples, the model trained under SPA provides more detailed and informative responses to various questions.

Prompt:
Who is Larry Page?

SFT response:
Larry Page is the co-founder and CEO of Google, a multinational technology company that provides various services such as search engine, online advertising, and cloud computing. He was born on March 26, 1973, in East Lansing, Michigan, and graduated from the University of Michigan with a degree in computer engineering. Page and Sergey Brin founded Google in 1998, and since then, it has become one of the most successful and influential companies in the world.

DPO response:
Larry Page is an American computer scientist and entrepreneur who co-founded Google, one of the world's largest technology companies, with Sergey Brin in 1998. He is the CEO of Alphabet Inc., the parent company of Google, and has been instrumental in the development and growth of the company. Page is known for his innovative ideas and his focus on developing new technologies that can improve people's lives. He has been recognized for his contributions to the field of computer science and has received numerous awards and honors, including the Presidential Medal of Freedom in 2015.

SPA response:
Larry Page is an American entrepreneur, computer scientist, and co-founder of Alphabet Inc., the parent company of Google, one of the world's largest technology conglomerates. Born on March 26, 1973, in East Lansing, Michigan, Page grew up in a family of academics and engineers. He earned his Bachelor's degree in Computer Engineering and Master's degree in Computer Science from Stanford University in 1995 and 1998, respectively.

In 1998, Page teamed up with Sergey Brin, his classmate at Stanford, to develop a new search engine called "Backrub," which later evolved into Google. Initially funded by venture capitalists, Google quickly gained popularity for its innovative search algorithm, user-friendly interface, and extensive collection of online content. Page served as the CEO of Google from its inception in 1999 until 2019, overseeing its rapid growth, expansion into new markets, and development of groundbreaking products and services, such as Android, Maps, YouTube, and Cloud Computing.

Under Page's leadership, Google became a dominant player in the tech industry, with a market capitalization of over $1 trillion and a global workforce of over 135,000 employees. In 2019, Page stepped down as CEO of Google and assumed the role of CEO of Alphabet, the holding company that reorganized Google's core businesses and diversified into new areas, including healthcare, artificial intelligence, and sustainability.

Aside from his entrepreneurial achievements, Page is known for his philanthropic efforts, particularly through the Larry Page Foundation, which supports research in renewable energy, education, and medical innovation. He has also been recognized for his contributions to the tech industry, receiving numerous awards, honors, and recognitions, including the Presidential Medal of Freedom, Time Person of the Year, and Forbes' Person of the Year.

Figure 6: **Example of responses in AlpacaEval 2.0.**

Prompt:
What causes the northern lights

SFT response:
The northern lights, also known as aurora borealis, are caused by the interaction between charged particles from the sun and the Earth's magnetic field. When the charged particles from the sun enter the Earth's atmosphere, they collide with the gases in the atmosphere, causing them to emit light. The colors of the northern lights are determined by the type of gas that is colliding with the charged particles. Green is the most common color, but other colors such as red, yellow, and blue can also be seen.

DPO response:
The northern lights, also known as aurora borealis, are caused by the interaction between charged particles from the sun and the Earth's magnetic field and atmosphere.

When the sun emits charged particles, such as electrons and protons, they travel through space towards the Earth. As they approach the Earth, they are deflected by the Earth's magnetic field, which creates a protective bubble around the planet.

When these charged particles enter the Earth's atmosphere, they collide with the gases in the atmosphere, such as oxygen and nitrogen, causing them to become excited and emit light. This light is what we see as the northern lights.

The colors of the northern lights depend on the type of gas that is being excited. For example, oxygen produces green and yellow lights, while nitrogen produces blue and red lights.

The intensity and frequency of the northern lights depend on the activity of the sun, which can vary over time. During periods of high solar activity, the northern lights can be seen more frequently and with greater intensity.

SPA response:
The northern lights, also known as aurora borealis, are a natural phenomenon caused by the interaction between charged particles from space and the Earth's magnetic field and atmosphere. The primary cause of the northern lights is the solar wind, a stream of charged particles (primarily electrons, protons, and alpha particles) emitted from the Sun's corona and solar atmosphere.

As the solar wind approaches the Earth, it encounters the magnetic field, which deflects and traps some of the charged particles in a region known as the magnetosphere. The magnetosphere extends outward from the Earth's surface to a distance of several hundred thousand kilometers and is shaped like a teardrop, with the magnetic field lines converging towards the polar regions.

When energetic particles from the solar wind enter the magnetosphere, they collide with the atoms and molecules in the Earth's upper atmosphere, primarily in the polar regions, where the magnetic field lines are strongest. These collisions ionize the atmospheric gases, creating a plasma region known as the ionosphere.

The ionization process releases energy in the form of electromagnetic radiation, including ultraviolet, visible, and radio waves. The visible auroral emissions occur when the energetic particles excite the electrons in the ionized gases, causing them to emit photons with specific wavelengths, resulting in the characteristic colors of the northern lights.

The colors of the aurora depend on the type of gas molecule involved in the collisions and the energy level of the excited states. Green and yellow auroras are typically associated with oxygen atoms, while red and blue auroras are caused by nitrogen molecules. Other gases, such as hydrogen, helium, and carbon dioxide, can also contribute to the auroral spectrum, resulting in additional colors like purple, pink, and white.

The intensity and frequency of auroral displays vary depending on solar activity levels, geomagnetic conditions, and seasonal factors. Solar maximum periods, when the Sun's activity is at its highest, tend to produce more frequent and intense auroral displays in both hemispheres. However, even during solar minimum periods, when solar activity is low, occasional auroral events can still occur in the polar regions.

Figure 7: **Example of responses in AlpacaEval 2.0.**

毕业论文（设计）文献综述和开题报告考核

导师对开题报告、外文翻译和文献综述的评语及成绩评定：

成绩比例	文献综述 占（10%）	开题报告 占（15%）	外文翻译 占（5%）
分值			

导师签名 ___________

年　月　日

学院盲审专家对开题报告、外文翻译和文献综述的评语及成绩评定：

成绩比例	文献综述 占（10%）	开题报告 占（15%）	外文翻译 占（5%）
分值			

开题报告审核负责人（签名/签章）___________

年　月　日