Personalized Adaptation via In-Context Preference Learning

Allison Lau
University of Toronto
allison.lau@mail.utoronto.ca

Younwoo (Ethan) Choi*
University of Toronto
ywchoi@cs.toronto.edu

Vahid Balazadeh*
University of Toronto
vahid@cs.toronto.edu

Keertana Chidambaram*
Stanford University
vck@stanford.edu

Vasilis Syrgkanis
Stanford University
vsyrgk@stanford.edu

Rahul G. Krishnan
University of Toronto
rahulgk@cs.toronto.edu

Abstract

Reinforcement Learning from Human Feedback (RLHF) is widely used to align Language Models (LMs) with human preferences. However, existing approaches often neglect individual user preferences, leading to suboptimal personalization. We present the Preference Pretrained Transformer (PPT), a novel approach for adaptive personalization using online user feedback. PPT leverages the in-context learning capabilities of transformers to dynamically adapt to individual preferences. Our approach consists of two phases: (1) an offline phase where we train a single policy model using a history-dependent loss function, and (2) an online phase where the model adapts to user preferences through in-context learning. We demonstrate PPT's effectiveness in a contextual bandit setting, showing that it achieves personalized adaptation superior to existing methods while significantly reducing the computational costs. Our results suggest the potential of in-context learning for scalable and efficient personalization in large language models.

1 Introduction

Reinforcement Learning from Human Feedback (RLHF) has emerged as a powerful technique for aligning large language models (LLMs) with human preferences, enabling them to generate higher-quality and more desirable outputs [1, 2, 3, 4]. However, standard RLHF aims to learn a general policy optimized for the entire population, often neglecting the diversity of individual preferences and potentially marginalizing specific groups [5, 6, 7, 8, 9]. This limitation can lead to suboptimal experiences for users whose preferences deviate from the majority.

Several approaches have been proposed to account for the diverse preferences of different populations. One strategy is to use multi-objective reinforcement learning [10, 11] techniques to train multiple interpolated proxy rewards and their corresponding optimal policies during the training phase. Then, they choose the policy that maximizes each new user's reward as it becomes known during the selection phase [12, 13]. However, as mentioned in Ramé et al. [6], those approaches require maintaining a large set of networks, potentially one for each possible preference profile. To

*Equal contribution with random ordering.

NeurIPS 2024 Workshop on Adaptive Foundation Models.

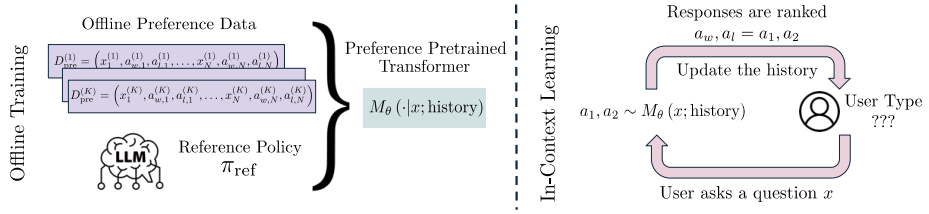


Figure 1: Preference Pretrained Transformer: (i) In the offline phase, we train a single policy to predict the preferred answers given the history of previous responses. (ii) In the online phase, the pretrained model interacts with the user, appends the interaction history to its context and generates more personalized responses.

address this issue, Ramé et al. [6] and Jang et al. [7] propose "rewarded soups" and "personalized soups," respectively, which learn separate reward and policy models for each preference criterion and aggregate the learned rewards or policies for new users. While effective, these approaches can be computationally expensive, particularly when the number of criteria is large, and may require retraining an entire model for every new criterion. Moreover, these methods often rely on a separate selection phase to identify user's relevant reward/policy model (e.g., by optimizing the interpolating coefficients of the trained models), which translates to an extra computation step for every new user.

To address these challenges, we introduce a novel framework for online personalized adaptation without the need for training separate models of each preference criterion. Our intuition is that a *history-dependent* policy should be able to identify the preference profile of new users after a few interactions with them. To learn such history-dependent policies, we leverage the in-context learning capabilities of transformer architectures [14, 15, 16], which has been recently proved to be effective in reinforcement learning and bandit problems by methods like Decision Pretrained Transformers (DPT) [17]. We call our model Preference Pretrained Transformer or PPT for short. PPT is two-fold: (i) During the *offline* phase, we employ a history-dependent loss function to train a *single* policy model that predicts the preferred responses given the history of responses within each preference criterion. In particular, we follow a direct preference optimization (DPO) approach to avoid learning a separate reward model [18]. (ii) During the *online* inference phase, for each new user, we follow an in-context learning approach by generating two potential responses for each prompt the user gives and asking the user to rank them. We then append those interactions to the trained model's context and continue the inference. This procedure allows the model to dynamically adapt to individual user preferences as the interaction progresses, rather than relying on a distinct validation set for model selection, as in prior work. See Figure 1 for an illustration of PPT.

We demonstrate the effectiveness of our approach in a contextual bandit setting, considering each user prompt as a unique context. The model's task is to generate a preferred response, which acts as the optimal action dependent on both the context and the user's preference group. This framework aligns with the contextual bandit setting, where an agent must choose the best action based on the current context to maximize the expected reward. We show its ability to achieve personalized in-context learning by outperforming the Personalized Soups baseline while reducing the computation cost by only learning one policy model. Our results suggest the potential of in-context learning for scalable and efficient personalization in LLMs.

2 Preference Pretrained Transformer

Consider a multi-preference learning setting, with a set of questions (prompts or contexts) $\mathcal{X}$, potential responses $\mathcal{A}$, and preference groups (or preference criteria) denoted by $\mathcal{G} = \{1, 2, \ldots, K\}$ for K number of groups. For each group $g \in \mathcal{G}$, assume an unknown function $r_g : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$, which assigns a reward value $r_g(a \mid x)$ to a potential response a given a question x and a preference group g. Moreover, assume there exist noisily rational annotators corresponding to different preference groups. An annotator from preference group g, prefers a response a_w over a_l for a question x following a Bradley–Terry model [19, 20]:

$$\mathbb{P}(a_w, a_l \mid x\,;\, g) = \frac{\exp\{r_g(a_w \mid x)\}}{\exp\{r_g(a_w \mid x)\} + \exp\{r_g(a_l \mid x)\}},$$

where in our notation, given an option between a_w and a_l, a_w was chosen by the user. Finally, we do not directly observe the reward values but only the offline preference data as a proxy:

Offline preference data. We assume the given offline datasets are stratified by the preference groups, i.e.,

$$D_{\text{pre}} = \left\{ D_{\text{pre}}^{(g)} \triangleq \left(x_1^{(g)}, a_{w,1}^{(g)}, a_{l,1}^{(g)}, x_2^{(g)}, a_{w,2}^{(g)}, a_{l,2}^{(g)}, \ldots, x_N^{(g)}, a_{w,N}^{(g)}, a_{l,N}^{(g)} \right) \right\}_{g=1}^{K}.$$

In this notation, each superscript $^{(g)}$ corresponds to a fixed preference group $g \in \mathcal{G}$, and each subscript $_i$ refers to a different question x_i and its preference data $a_{w,i}, a_{l,i}$ for N questions. [2]

User preference profile. As discussed, we assume separate offline preference data for each preference group. However, we also allow for any preference profile induced by rewards in the convex hull of the true reward models of each preference group. Concretely, we denote a user preference profile by $z = (\alpha_1, \alpha_2, \ldots, \alpha_K)$ for $\alpha_1, \ldots, \alpha_K \geq 0$ and $\sum_{g=1}^{K} \alpha_g = 1$. Then, the corresponding reward function is defined as a linear combination of multiple reward functions, i.e., $r_z(a \mid x) = \sum_{g=1}^{K} \alpha_i \cdot r_g(a \mid x)$. We denote the set of all user preference profiles by $\Delta(\mathcal{G})$.

Online phase. Given the offline datasets $D_{\text{pre}} \in \mathcal{D}$, our goal is to train an *in-context learning* model $M : \mathcal{X} \times \mathcal{D} \to \Delta(\mathcal{A})$ that can generate *personalized* responses during the online deployment phase. The model interacts with a fixed user $z^\star \in \Delta(\mathcal{G})$, which is unknown to the model, for iterations $j = 1, 2, \ldots$ as follows:

1. The user (with profile $z^\star$) asks a question x_j from the model.
2. The model generates two potential responses given the question and the history of interactions, i.e., $a_{w,j}, a_{l,j} \sim M(x_j \,;\, x_1, a_{w,1}, a_{l,1}, \ldots, x_{j-1}, a_{w,j-1}, a_{l,j-1})$ or $a_{w,j}, a_{l,j} \sim M(x_j)$ if $j = 1$.
3. The user prefers $a_{w,j}$ over $a_{l,j}$ based on $\mathbb{P}(a_{w,j}, a_{l,j} \mid x_j \,;\, z^\star)$.

Preference Pretrained Transformer. Here, we present and motivate our proposed approach. We assume the questions in the offline and online datasets come from an unknown distribution $\mathcal{D}_{\mathcal{X}}$. Given a model M_θ and a preference profile $z^\star$, denote a history of interactions between the model and the user up to time h as $H_h = (x_j, a_{w,j}, a_{l,j})_{j=1}^{h}$ with $H_0 = \emptyset$. The joint distribution of the history is given as follows:

$$\mathcal{D}_{\mathcal{H}}(H_h \,;\, M_\theta, z^\star) = \prod_{j=1}^{h} \mathcal{D}_{\mathcal{X}}(x_j) \cdot M_\theta(x_j \,;\, H_{j-1})\,(a_{w,j}) \cdot M_\theta(x_j \,;\, H_{j-1})\,(a_{l,j}) \cdot \mathbb{P}(a_{w,j}, a_{l,j} \mid x_j \,;\, z^\star)\,.$$

We define the notion of *optimal in-context learning models* as the ones maximizing the following:

$$\max_{M_\theta} \mathbb{E}_{x \sim \mathcal{D}_{\mathcal{X}}, H \sim \mathcal{D}_{\mathcal{H}}(\cdot \,;\, M_\theta, z^\star), a \sim M_\theta(x \,;\, H)} [r_{z^\star}(a \mid x)] - \beta \cdot \mathrm{KL}\left(M_\theta(x \,;\, H) \parallel \pi_{\text{ref}}(\cdot \mid x)\right). \tag{1}$$

This objective function trades off between exploration and exploitation by maximizing the reward of the generated responses while allowing the model full control over the previously generated answers in history. Moreover, β is a parameter controlling the deviation from some reference policy $\pi_{\text{ref}} : \mathcal{X} \to \Delta(\mathcal{A})$. However, exact optimization of the objective function (1) is challenging since (i) the history (in-context data) H in the expectation also depends on the learnable model M_θ, and (ii) $z^\star$ is unknown. Here, we propose an approximation to this objective in two ways. First, we replace the model M_θ with the reference policy π_{ref} in the history distribution $\mathcal{D}_H$. Such approximation is reasonable since the Lagrangian in (1) already requires the learned model M_θ to be within a KL-ball from the reference policy. Second, we assume a uniform distribution $\texttt{Unif}(K)$ for $z^\star$ over different preference groups, i.e., $\mathbb{P}(z^\star = g) = \frac{1}{K}$ for all $g \in \mathcal{G}$. While this assumption does not consider the users inside the reward convex hull, our empirical results demonstrate competitive performance for such users. Applying these two approximations, we get the following objective function:

$$\max_{M_\theta} \mathbb{E}_{x \sim \mathcal{D}_{\mathcal{X}}, z^\star \sim \text{Unif}(K), H \sim \mathcal{D}_{\mathcal{H}}(\cdot \,;\, \pi_{\text{ref}}, z^\star), a \sim M_\theta(x \,;\, H)} [r_{z^\star}(a \mid x)] - \beta \cdot \mathrm{KL}\left(M_\theta(x \,;\, H) \parallel \pi_{\text{ref}}(\cdot \mid x)\right).$$

Noting that $D_{\text{pre}}^{(g)} \sim \mathcal{D}_{\mathcal{H}}(\cdot \,;\, \pi_{\text{ref}}, g)$, the above can be simplified as

$$\max_{M_\theta} \frac{1}{K} \sum_{g=1}^{K} \left(\mathbb{E}_{x \sim \mathcal{D}_{\mathcal{X}}, H \sim D_{\text{pre}}^{(g)}, a \sim M_\theta(x \,;\, H)} [r_g(a \mid x)] - \beta \mathrm{KL}\left(M_\theta(x \,;\, H)\,(a) \parallel \pi_{\text{ref}}(a \mid x)\right) \right).$$

[2] The setting is similar to a contextual bandit rather than a Markov decision process.

Instead of learning a reward model and optimizing the above objective, we follow a direct preference optimization (DPO) approach that results in the following loss function [18]:

$$\mathcal{L}(\theta) = -\frac{1}{K}\sum_{g=1}^{K}\mathbb{E}_{\{(x,a_w,a_l),H\}\sim D_{\text{pre}}^{(g)}}\left[\log\sigma\left(\beta\log\frac{M_\theta(x\,;\,H)\,(a_w)}{\pi_{\text{ref}}(a_w\mid x)} - \beta\log\frac{M_\theta(x\,;\,H)\,(a_l)}{\pi_{\text{ref}}(a_l\mid x)}\right)\right]. \quad (2)$$

3 Experiments

LLMs can be thought of as a contextual bandit, where the prompt can be represented as the context x. Then, each response corresponds to an arm of the bandit $a \in \mathcal{A}$. The reward for each response then is a function of both the prompt x as well as the chosen arm a. For group z let this reward function be represented as $r_z(a, x)$. The preferences between arms are generates through the Bradley-Terry model as described before [19, 20]. In this preliminary version, we use this framework to run proof-of-concept experiments for our algorithm.

3.1 Experimental Setup

We consider a simplified scenario involving three subpopulations, where each context vector x represents a question and each action $a \in \{0, 1, 2, 3\}$ represents one of four possible responses. To create the preference dataset, we uniformly sample N_c context vectors from the unit hypercube $[0, 1]^3$. All the contexts are annotated by users from the first subpopulation, 80% are annotated by users from the second subpopulation, and 60% by users from the third subpopulation. For the annotation process, we generate two candidate actions a', a'' from a uniformly random reference policy $\pi_{\text{SFT}} = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ and rank the two actions using the subpopulation-specific reward models. The reward model is formulated as a linear function, similar to the linear contextual bandit setup:

$$r_z(a, x) = f_\phi(x)^\top \theta_z(a) + \mathcal{N}(0, \sigma)$$

where $f_\phi : \mathbb{R}^3 \to \mathbb{R}^4$ is a context encoding function, θ_z is a reward matrix for subpopulation z and $\mathcal{N}(0, \sigma)$ is a Gaussian noise with mean 0 and standard deviation $\sigma = 0.01$. We define θ_z such that each column $\theta_z(j)$ corresponds to the rewards obtained by selecting action j for a user from group z. We have $\theta_i(j) = [\theta_{i1}(j), \theta_{i2}(j), \theta_{i3}(j), \theta_{i4}(j)]$, where $\theta_{i1}(j) = \theta_{i2}(j) = \theta_{i3}(j) = \theta_{i4}(j) = r_{ij}$ for any group i and action j, we choose $r_{ij} \in \{1, 3, 5, 7\}$ to ensure that the reward model is deterministic and contrasting (i.e., there is a unique ranking of actions for each group.) We choose the context encoding function f_ϕ to be a linear function. We then train a simple transformer using history-dependent DPO on $\widetilde{D}_{\text{pre}} = \{(x_t, a_t, \bar{a}_t)\}_{j=1}^{T}$ where $T = 15$, with the loss function (2). For our baseline Personalized Soups (PS) [7], we trained 3 transformers, one for each subpopulation, using standard DPO on data specific to each subpopulation. In both cases, the transformers consist of 6 layers, 4 attention heads, and a hidden dimension of 256.

3.2 Results

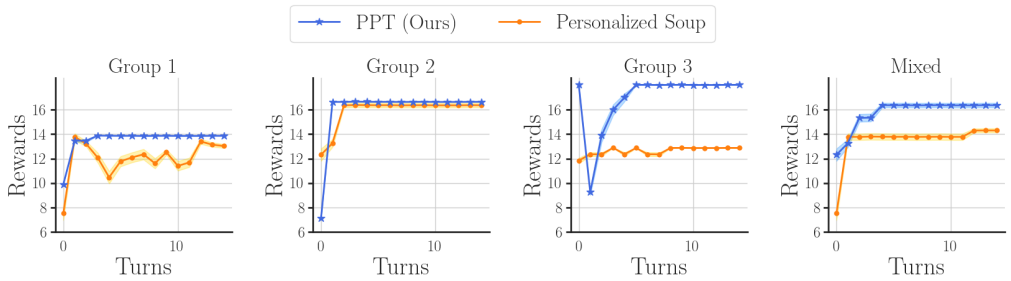
We sample $L = 50$ test contexts $\{x_i^{\text{test}}\}_{i=1}^{50}$ from the training set and evaluate the rewards obtained at each turn for $T = 15$ turns, for users from each group and for a user that has mixed reward function $r_{\tilde{z}}(a, x) = \boldsymbol{\lambda}^\top r(a, x)$ where $\boldsymbol{\lambda} \in \mathbb{R}^3$ is sampled randomly from the 3-simplex and $r(a, x) = [r_1(a, x), r_2(a, x), r_3(a, x)]$.

We evaluate PPT against the baseline which is Personalized Soups (PS) [7]. For PS, we begin by constructing 100 interpolated models by performing weighted linear interpolation of the three pretrained transformers. We initialize the best model $M^\star$ by randomly sampling one from 100 interpolated models. For each turn t, we evaluate the mean accuracy of $M^\star$ in predicting the preferred action given $\{x_i^{\text{test}}\}_{i=1}^{L}$. We then sample a random context x_j^{val} and generate two actions a', a'' from a uniform reference policy. Then we select the winning action denoted a_w based on the test user's preferences. For each interpolated model, we compute the log-probability of a_w for the context x^{val} and add to the model's cumulative score. $M^\star$ is then updated based on the accumulated score.

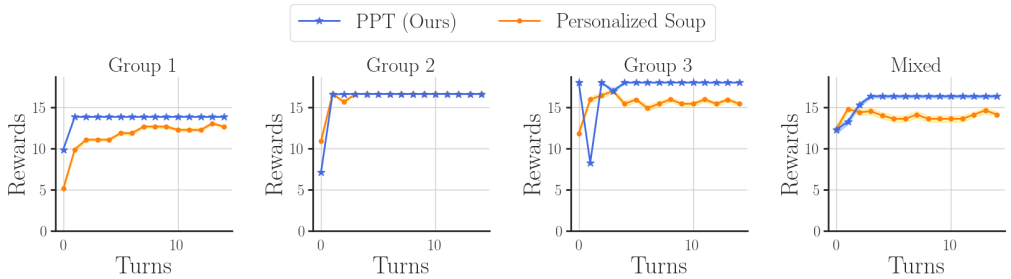
For our model, we maintain a history of interactions H for T turns. We use the same set of validation contexts $\{x_j^{\text{val}}\}_{j=1}^{T}$ and test contexts $\{x_i^{\text{test}}\}_{i=1}^{L}$. For each turn t, we sample responses from $M_\theta(\cdot|x_i^{\text{test}}, H)$ and similarly evaluate the accuracy of M_θ in predicting the preferred action on

$\{x_i^{\text{test}}\}_{i=1}^{L}$. We then generate two actions a', a'' from $M_\theta(\cdot|x_j^{\text{val}}, H)$ and select a_w based on the test user's preferences. The history is then updated with $H \leftarrow H \cup \{(x_j^{\text{val}}, a_j, \bar{a}_j)\}$.

Figure 2 shows that our method consistently outperforms PS across all user groups. The rewards increase with the number of turns. In contrast, PS shows some fluctuations in rewards even with the large number of interpolated models and turns. This highlights the advantage of our approach in not requiring separate models for each subpopulation; instead, we train a single model capable of adapting to any user preference. Furthermore, the consistent improvement in rewards across different groups, including users with mixed preferences, demonstrates the robustness of our method. As the number of turns grows, our model becomes increasingly accurate in predicting the user's preferred actions, even if it does not perform optimally in the initial iterations. This behavior underscores PPT's ability to effectively learn in-context, dynamically adapting without the need for retraining or complex model selection procedures.



(a) Rewards vs Turns ($N_c = 500$)



(b) Rewards vs Turns ($N_c = 1000$)

Figure 2: Comparison of rewards between PPT (ours) and the Personalized Soups (PS) over 15 interaction turns for different user groups. Figure 2a and Figure 2b show results with $N_c = 500$ and $N_c = 1000$ context vectors, respectively. Each subplot corresponds to tests with users from one of the three subpopulations and a user with mixed preferences. The results demonstrate that PPT consistently outperforms the PS baseline across all groups. The increase in rewards for our method as the number of turns grows indicates effective in-context learning and dynamic adaptation to user preferences.

4 Conclusion

In this paper, we introduced the Preference Pretrained Transformer (PPT), a novel framework for online personalized adaptation of language models to user preferences. PPT addresses the limitations of existing RLHF approaches by leveraging the in-context learning capabilities of transformers to dynamically adapt to individual user preferences. We demonstrated the effectiveness of PPT in a contextual bandit setting, showing that our model achieves personalized adaptation superior to the Personalized Soups baseline while significantly reducing computational costs. By training a single policy model instead of multiple preference-specific models, PPT offers a more scalable and efficient solution for LM personalization. A key limitation of this work is the lack of experiments with large language models, which will be crucial for validating the approach's scalability and real-world applicability. Future work should explore the application of PPT to more complex language tasks and investigate its performance with larger language models and diverse user populations.

References

[1] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

[2] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

[3] 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*, 2022.

[4] 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. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[5] Peter Vamplew, Richard Dazeley, Cameron Foale, Sally Firmin, and Jane Mummery. Human-aligned artificial intelligence is a multiobjective problem. *Ethics and information technology*, 20:27–40, 2018.

[6] Alexandre Ramé, Guillaume Couairon, Mustafa Shukor, Corentin Dancette, Jean-Baptiste Gaya, Laure Soulier, and Matthieu Cord. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards, 2023. URL https://arxiv.org/abs/2306.04488.

[7] Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong Wang, Jack Hessel, Luke Zettlemoyer, Hannaneh Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu. Personalized soups: Personalized large language model alignment via post-hoc parameter merging, 2023. URL https://arxiv.org/abs/2310.11564.

[8] Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*, 2023.

[9] Souradip Chakraborty, Jiahao Qiu, Hui Yuan, Alec Koppel, Furong Huang, Dinesh Manocha, Amrit Singh Bedi, and Mengdi Wang. Maxmin-rlhf: Towards equitable alignment of large language models with diverse human preferences, 2024. URL https://arxiv.org/abs/2402.08925.

[10] Leon Barrett and Srini Narayanan. Learning all optimal policies with multiple criteria. In *Proceedings of the 25th international conference on Machine learning*, pages 41–47, 2008.

[11] Kaiwen Li, Tao Zhang, and Rui Wang. Deep reinforcement learning for multiobjective optimization. *IEEE transactions on cybernetics*, 51(6):3103–3114, 2020.

[12] Daniel Marta, Simon Holk, Christian Pek, Jana Tumova, and Iolanda Leite. Aligning human preferences with baseline objectives in reinforcement learning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7562–7568. IEEE, 2023.

[13] Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. Fine-grained human feedback gives better rewards for language model training. *Advances in Neural Information Processing Systems*, 36, 2024.

[14] Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022.

[15] Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second. *arXiv preprint arXiv:2207.01848*, 2022.

[16] Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by

gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR, 2023.

[17] Jonathan N. Lee, Annie Xie, Aldo Pacchiano, Yash Chandak, Chelsea Finn, Ofir Nachum, and Emma Brunskill. Supervised pretraining can learn in-context reinforcement learning, 2023. URL https://arxiv.org/abs/2306.14892.

[18] 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 *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[19] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

[20] 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.