# FACTUAL AND PERSONALIZED RECOMMENDATIONS USING LANGUAGE MODELS AND REINFORCEMENT LEARNING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Recommender systems (RSs) play a central role in connecting users to content, products, and services, matching candidate items to users based on their preferences. While traditional RSs rely on implicit user feedback signals, conversational RSs interact with users in natural language. In this work, we develop a *comPelling*, *Precise*, *Personalized*, *Preference-relevant* language model (P$^4$LM) that recommends items to users while putting emphasis on explaining item characteristics and their relevance. P$^4$LM uses the *embedding space* representation of a user's preferences to generate compelling responses that are factually-grounded and relevant w.r.t. the user's preferences. Moreover, we develop a joint reward function that measures precision, appeal, and personalization, which we use as AI-based feedback in a reinforcement learning-based language model framework. Using the MovieLens 25M dataset, we demonstrate that P$^4$LM delivers compelling, personalized movie narratives to users.

## 1 INTRODUCTION

Recommender systems (RSs) have emerged as a dominant way in which users discover content, products, and services (Resnick & Varian, 1997). Traditional RSs match candidate items to users based on their estimates for items preferences, possibly conditioned on some query or context. However, these preferences are often based on implicit user behavioral signals, such as clicks, number of watches, ratings, purchases, etc. Unfortunately, these provide little opportunity for an RS to elicit high-bandwidth preference information from users, explain recommendations, or for users to critique and steer their interaction with the RS. *Conversational RSs* have therefore attracted considerable attention as means to use natural-language interaction to facilitate more effective communication between RSs and their users (Sun & Zhang, 2018; Lei et al., 2020; Shen et al., 2023).

The emergence of language models (LMs) as a powerful paradigm for user engagement (Li et al., 2018; Friedman et al., 2023) suggests their use as a vehicle for conversational RSs. However, this requires LMs to engage in a personalized manner, adhering to users' preferences. In this paper, we explore the intersection of RSs and LMs, and more particularly, the use of LMs to enrich the user experience in RSs. We develop techniques which allow an LM to communicate the nuances of recommended items to a user, detailing their features, benefits, and explaining their *alignment with a user's preferences*. Such *personalized LMs* are not meant to "convince" users in the traditional sense, but rather, to articulate the *genuine and relevant merits* of a recommended item relative to the user.

Personalized LMs offer users a fully tailored RS experience, ensuring they find what they truly need and value. However, a number of challenges must be addressed in this endeavor: (i) any recommended item should be predicted to have maximal value given the user's preferences; (ii) the integrity and accuracy of an item's information is paramount; (iii) the personalized LM should present a reasonably comprehensive portrayal of the item by describing its merits and drawbacks, with a focus on *relevance* to the user's preferences; (iv) and finally, the LM's explanations or endorsements should be compelling and appealing to the user, provided that it meets the other criteria. In this work, we develop a framework centered around these four principles.

A key question we addressed in this work is how to effectively utilize the information captured by an RS embedding space to generate a factual, personalized, compelling, and relevant recommendations.

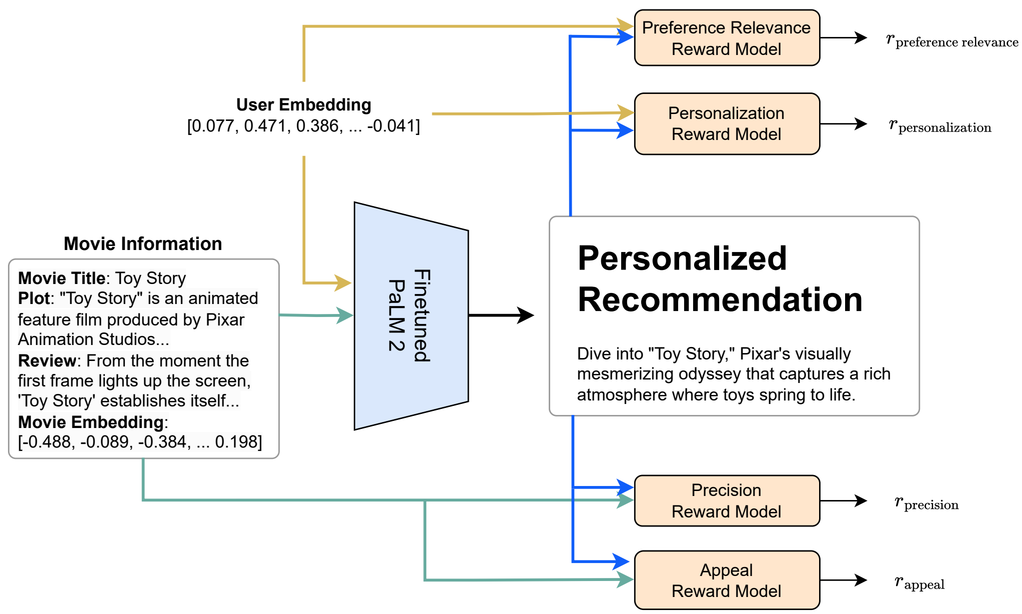



Figure 1: The P$^4$LM Learning Framework for Recommendation Endorsement Generations.

Our contributions are three-fold. First, we quantify the aforementioned four attributes using reward functions, enabling systematic evaluation. Second, leveraging recent advances in reinforcement learning from AI feedback (RLAIF) (Lee et al., 2023), we develop an LM fine-tuning methodology to better align with these four rewards (see Figure 1 for the schematic diagram illustrating the RLAIF framework). Our developed model, which we term P$^4$LM, not only comprises semantic skills, but also understands users' preferences encoded in the RS embedding space, providing factual, compelling, personalized endorsements. Finally, building on the MovieLens 25M dataset (Harper & Konstan, 2015) we showcase the potential of P$^4$LM , powering a conversational movie recommender that promotes customized, relevant, and holistic interactions for users.

We begin with a brief introduction of RSs, LMs and the use of contextual Markov decision processes (CoMDPs) for modeling generative language problems of RSs (Section 2). We then describe the four principles, (i.e., personalization, precision, appeal, and preference relevance), which we incorporate into training of LMs for RSs (Section 3), followed by an reinforcement learning based fine-tuning methodology for training P$^4$LM (Sections 4). Finally, we demonstrate the effectiveness of P$^4$LM in generating factual, personalized, and compelling movie endorsement narratives for users within the MovieLens 25M benchmark dataset (Section 5).

## 2 PRELIMINARIES

In this section we present some basic background, outline our problem formulation, and establish the terminology used throughout the paper.

**Recommender Systems (RSs).** To model user-item behavioral relationships in a personalized RS, we assume a standard collaborative filtering (CF) task (Su & Khoshgoftaar, 2009). Collaborative filtering finds similar patterns among users, filtering out items based on ratings of similar users. Given a user $u \in \mathcal{U}$, we use $r_{u,i}$ (e.g., 1–5 stars) to denote the rating of item $i \in \mathcal{I}$ by user $u$. Let $\mathcal{R}$ denote the $|\mathcal{I}| \times |\mathcal{U}|$ (usually sparse) ratings matrix corresponding to the *ratings dataset* $\mathcal{R} = \{(u, i, r_{u,i}) : r_{u,i} \neq 0\}$. To predict users' preference behavior, an RS learns user and item representations from the ratings dataset $\mathcal{R}$ using a CF approach. Then, the resulting *item embedding* maps each item $i$ to a vector representation $\mathbf{i}$ of its (latent) attributes. Note that these embeddings are typically not interpretable. Similarly, user preferences are captured by a *user embedding*, mapping users $u$ to a vector representation $\mathbf{u}$.

Methods including matrix factorization (Mnih & Salakhutdinov, 2007) or neural CF (Rendle et al., 2020; He et al., 2017; Beutel et al., 2018) are used to learn the user and item embeddings, which assumes a *two-tower model* (or *dual encoder*) in which users and items are passed through separate (but co-trained) deep neural nets (DNNs) to produce their respective vector embeddings $\mathbf{u}$ and $\mathbf{i}$. These are then combined via dot product to predict user-item affinity $\hat{r}_{i,u}$ (Yi et al., 2019; Yang et al., 2020). We view $\mathbf{i}$ as a (learned) latent feature vector characterizing item $i$ and $\mathbf{u}$ as parameterizing user $u$'s estimated *utility (or preference) function* over these features.

**Language Models (LMs).** In this work, we inject a user's behavioral information into a seq2seq LM (Vaswani et al., 2017) to generate personalized recommendation responses. We assume a dataset of the form $\mathcal{D} = \{(\mathbf{I}^{(k)}, \mathbf{i}^{(k)}, \mathbf{u}^{(k)}, Y^{(k)})\}_{k=1}^{|\mathcal{D}|}$, where $\mathbf{I}$ is a textual description of some item $i \in \mathcal{I}$ (e.g., descriptions, positive/negative reviews from different users); $\mathbf{i}$ is the CF embedding vector of $i$; $\mathbf{u}$ is the CF embedding vector of a user $u \in \mathcal{U}$; and finally, $Y$ is a textual response (e.g., compelling recommendation, endorsement or explanation) tailored to the user. We refer to Appendix C for details on the generation of $\mathcal{D}$.

Let $N_{\mathbf{I}}$ be an upper-bound on the length (number of tokens) of any item description $\mathbf{I}$.[1] The role of an LM is to predict the probability $\mathbb{P}\big(Y = \{y_n\}_{n=0}^{N-1} \mid y_0, \mathbf{I}, \mathbf{i}, \mathbf{u}\big)$ of the personalized response $Y$ ($N$ tokens), conditioned on the item description $(\mathbf{I}, \mathbf{i})$ and user embedding $\mathbf{u}$.

In standard LMs, a Transformer (Wolf et al., 2019) architecture $\mathrm{T}$ encodes an item's textual context $\mathbf{I}$ as an $N_{\mathbf{I}}$-length sequence of embeddings $(z_0, \ldots, z_{N_{\mathbf{I}}-1})$ induced by the transformer's attention layers. For convenience, we concatenate these into a single embedding $z \in \mathcal{Z} \subseteq \mathbb{R}^d$, where $d$ is the dimension of the latent space. The text response $Y = \{y_n\}_{n=0}^{N-1}$ is sampled token-by-token in an auto-regressive manner using a decoder $\Psi$; i.e., $Y \sim \Psi\big(\cdot \mid z\big) := \prod_{n=0}^{N-1} \Psi\big(y_n \mid y_0, \ldots, y_{n-1}; z\big)$, where $y_0$ is a fixed start-of-sentence token (Chien & Kuo, 2019). To incorporate behavioral information into the LM, the standard LM is augmented with adapters (Pfeiffer et al., 2020) $\mathrm{W}_I, \mathrm{W}_U : \mathcal{V} \mapsto \mathcal{Z}$, to induce the language model: $\Psi \circ (\mathrm{T} \times \mathrm{W}_I \times W_U)$ (Jaech & Ostendorf, 2018). Here, $\mathrm{T}$ maps text-input tokens to $\mathcal{Z}$ whereas $\mathrm{W}_I$ (resp., $\mathrm{W}_U$) maps item (resp., user) CF-embedding vectors $\mathcal{V}$ to $\mathcal{Z}$. Importantly, $\mathrm{T}$, $\mathrm{W}_I$, and $\mathrm{W}_U$ map tokens and CF vectors to a common space so that their relationship can be captured by the transformer's attention mechanism.

**Contextual Markov Decision Processes (CoMDPs).** CoMDPs have been used to model token-wise generative language problems (Li et al., 2016; Asadi & Williams, 2016; Jaques et al., 2019), and can also be used in conversational RSs. In this MDP, the LM acts as a policy which maps text inputs and user/item behavioral embedding vectors to generated responses.

Let $(\mathcal{C}, \mathcal{S}, \mathcal{A}, P, r, s_0, N)$ denote the CoMDP, where the observable context space $\mathcal{C}$ contains item/user information $\mathcal{I}$, $\mathbf{i}$ and $\mathbf{u}$. The horizon $N$ is the length of the generated text. The state space $\mathcal{S}$ at the $n$-th turn ($n < N$) is the sequence of tokens $\{y_0, \ldots, y_{n-1}\}$ generated thus far, with $s_0$ being the start-of-sentence token $y_0$. The action space $\mathcal{A}$ is the language token vocabulary, with action $a \in \mathcal{A}$ representing any possible next token. The transition kernel $P$ models the next token distribution given the current sequence and contexts, which coincides with the LM policy (and is thus known). Finally, the reward function $r$ measures the overall quality of the generated text. Our goal is to find a policy $\pi^*$ which achieves maximum expected cumulative return, i.e., $\pi^* \in \arg\max_\pi J_\pi := \mathbb{E}[\sum_{n=0}^{N-1} r_t \mid P, s_0, \mathcal{C}, \pi]$. Note that the size of the tokenized state and action spaces grow exponentially with the vocabulary size.

## 3 Factual & Personalized Recommendations with LMs

A key question when using LMs for recommendation is how to effectively use the information captured by the RS embedding space to generate a factual, personalized, compelling, and relevant text response. Treating an LM as a factored distribution of item-user information over generated text tokens, one standard approach is to learn this model with *behavioral cloning (BC)* (Sasaki &

[1] If the actual description $\mathbf{I}$ has fewer tokens than $N_{\mathbf{I}}$, remaining spaces in the utterance will be padded by a specific token and masked.

Yamashina, 2020), by maximizing the conditional log-likelihood w.r.t. to the dataset $\mathcal{D}$:

$$\min_{\Psi} \; L_{\text{Cond}}(\Psi) := -\mathbb{E}_{(\mathbf{I},\mathbf{i},\mathbf{u},Y)\sim D}\Big[\sum_{n=0}^{N-1} \log \Psi(y_n \mid y_0, \ldots, y_{n-1}; \mathbf{I}, \mathbf{i}, \mathbf{u})\Big].$$

While this model may learn to interpret the behavioral information captured in the RS embeddings, the LM might actually lean towards disregarding the embedding contexts due to the typically more predictable nature of token generation when given text inputs. Consequently, the model might concentrate solely on text information, effectively degenerating to a non-contextual LM. To prevent this from occurring, and more importantly to ensure the LM can offer a comprehensive RS experience, we incorporate four key metrics into our training procedure; namely, *personalization*, *precision*, *appeal*, and *preference relevance*. We detail these next.

**Precision.** LM-based personalized recommendation can be viewed as a special form of abstractive summarization (Zhang et al., 2020a; Liu et al., 2022): the generated text should capture item characteristics that explain why a user would benefit from the recommendation. To preserve the RS's integrity, of course, one must emphasize *truthfulness* in its recommendation. That is, the RS's generated recommendation should describes genuine merits (and drawbacks) of the item, rather than persuasive distortions.

While recent summarization techniques produce highly coherent texts, they often suffer from *hallucinations* (Ji et al., 2023) – the tendency to generate information unsupported by the input text. Such factual inconsistencies may therefore limit their real-world applicability. Inspired by Roit et al. (2023) and Honovich et al. (2022), we evaluate factuality in our LM-based RS using an *entailment reward* (Bowman et al., 2015). Unlike widely-used metrics, such as ROUGE (Lin, 2004), that are ineffective at hallucination detection, we adopt a *textual entailment* (or natural language inference (NLI)) metric to measure truthfulness of our generated text, viewing it as a partial summary of an items's description. Particularly, given a description $\mathbf{I}$, we define the NLI score $\text{NLI}(Y; \mathbf{I})$ of text-token sequence $Y$ as the probability of entailment under a classifier trained on several textual entailment datasets (see e.g., MacCartney & Manning (2007)). While this metric is not specifically tailored to summarization tasks, Honovich et al. (2021) show that it effectively detects factual inconsistencies in generated text. Since faithful summaries should be textually entailed by the input documents, such a metric provides informative feedback about the precision of generated item texts.

Of course, factual entailment is clearly insufficient in and of itself. In fact, it is rather easy to optimize a degenerate response which maximizes factual entailment (e.g., producing summaries that are highly extractive (Ladhak et al., 2021) or uninformative (Skalse et al., 2022)). In what follows we describe three other metrics we require for a comprehensive recommendation experience.

**Appeal.** Recent work has paid increasing attention to enriching recommendations to appeal to users (Felfernig et al., 2007; Zhang et al., 2020b). To the extent that we do not sacrifice user welfare, personalization, or factuality, such recommendations have value as they encourage users to accept recommendations of high personal utility. With recent LM technologies (Google et al., 2023; OpenAI, 2023), a plausible approach is to simply prompt an LM to generate an *endorsement* to complement its item recommendation. Such an endorsement, apart from being factual, should be compelling for the user. However, without systematic evaluation of such methods (e.g., do users find them appealing or compelling), it remains unclear whether they can improve the user experience. Quantifying appeal is challenging, as it may depend on subjective factors such as *style* (concise phrases over detailed explanations) and *language* (compelling, eloquent pitches over dry factual summaries).

To assess appeal, we use a dataset of pairwise human/machine demonstrations (see Appendix C for details on its construction). We develop an *appeal model* which scores the generated text $Y$ and assess how compelling it are, using learning from human/AI feedback (LAIF) (Christiano et al., 2017). Specifically, let $\mathcal{D}_{\text{app}} = \{(Y_w^{(k)}, Y_l^{(k)}; \mathbf{I})\}_{k=1}^{|\mathcal{D}_{\text{app}}|}$ be a labeled dataset reflecting the relative appeal of two recommendation texts $Y_w, Y_l$ given textual item description $\mathbf{I}$. Here, $Y_w \succ Y_l | \mathbf{I}$ indicates that $Y_w$ is more compelling given $\mathbf{I}$. Assuming these relationships are governed by a latent model $\text{App}(Y; \mathbf{I})$, we parameterize it via Bradley-Terry (Huang et al., 2006), where the appeal distribution is defined by

$$p_{\text{app}}(Y_w \succ Y_l; \mathbf{I}) = \frac{\exp(\text{App}(Y_l; \mathbf{I}))}{\exp(\text{App}(Y_w; \mathbf{I})) + \exp(\text{App}(Y_l; \mathbf{I}))}.$$

We estimate the parameters of the reward model via maximum likelihood by formulating the problem as a binary classification with a negative log-likelihood loss: $L_{\text{MLE}}(\text{App}, \mathcal{D}_{\text{app}}) =$

$-\mathbb{E}_{(Y_w, Y_l; \mathbf{I}) \sim \mathcal{D}_{\text{app}}} \log \sigma(\text{App}(Y_w; \mathbf{I}) - \text{App}(Y_l; \mathbf{I}))$. To reduce variance, we normalize this by subtracting the population mean so that $\mathbb{E}_{(Y, \mathbf{I}) \sim \mathcal{D}_{\text{app}}}[\text{App}(Y; \mathbf{I})] = 0$ for all contexts $\mathbf{I}$.

**Personalization.** A conversational RS is only effective to the extent that it recommends, and ultimately, the user accepts, items of significant value to the user. Thus, *personalization* is perhaps the foremost criterion with which to evaluate an LM-based RS. Particularly, we wish to evaluate the extent to which the LM's generated response $Y$ corresponds to an item with high utility for a user $u$. To this end, we develop a scoring model $\text{Per}(Y; \mathbf{i}, \mathbf{u})$ which interprets the semantics of text $Y$ to quantify its value as a personalized recommendation.

To achieve this, recall the dataset $\mathcal{D} = \{(\mathbf{I}^{(k)}, \mathbf{i}^{(k)}, \mathbf{u}^{(k)}, Y^{(k)})\}_{k=1}^{|\mathcal{D}|}$ of item description, item CF embedding vector, user CF embedding vector, and textual response tailored to the user, and the estimated utility that is the dot product $\hat{r} = \mathbf{i} \cdot \mathbf{u}$ of their CF embedding vectors. To measure personalization one could learn a reward model $\text{Per}(Y; \mathbf{i}, \mathbf{u})$ that predicts the utility $\hat{r}$ based on textual response $Y$. However, this approach relies on a strong assumption that such text alone is predictive of user-item utility. Alternatively, we can also employ the LAIF approach (Christiano et al., 2017) that leverages preference feedback to learn a personalization reward model. Using the same dataset $\mathcal{D}$, and assuming the recommendation text is more personalized than item description, i.e., $Y \succ \mathbf{I} | \mathbf{i}, \mathbf{u}$,[2] a Bradley-Terry based personalization reward model $\text{Per}(Y; \mathbf{i}, \mathbf{u})$ can be learned by minimizing the negative log-likelihood loss: $L_{\text{MLE}}(\text{Per}, \mathcal{D}_{\text{per}}) = -\mathbb{E}_{(Y, \mathbf{I}; \mathbf{i}, \mathbf{u}) \sim \mathcal{D}_{\text{per}}} \log \sigma(\text{Per}(Y; \mathbf{i}, \mathbf{u}) - \text{Per}(\mathbf{I}; \mathbf{i}, \mathbf{u}))$.

**Preference Relevance.** While appeal and personalization distinguish compelling recommendations for a user from simple factual item summaries, they do not capture the full *relevance* of the LM's response w.r.t. a user's preferences. For example, the LM might still describe item attributes that the user has no interest in (positively or negatively). To address this, we assume access to a *textual description* of a user's preferences (we later describe how we create these from user CF embeddings). We train an additional reward model, $\text{Prel}(Y; \mathbf{I}, \mathbf{u})$, which explicitly measures the semantic similarity between a user's description of preferences and the generated text, constrained to attributes of the recommended item. More specifically, we assume availability of a mapping from a user's CF embedding vector $\mathbf{u}$ to a textual description of their preferences. We train this mapping using a dataset of user embeddings and textual descriptions $\{\text{U}_j(\mathbf{u})\}_{j=1}^{J}$ (see Appendix C for details on the generation of this dataset).

Next, for each $(\mathbf{I}, \mathbf{u}, Y)$, we encode the user's textual preferences $\{\text{U}_j(\mathbf{u})\}_{j=1}^{J}$ and the item description $\mathbf{I}$ using an *LM semantic encoder*.[3] Then, we rank each textual preference using cosine similarity of its encoded counterpart and encoded item. This, in turn, determines which of the $J$ preference texts are most relevant to the item of interest. Finally, we use the same model to encode the recommendation response $Y$ and compute its cosine similarity with the user preference texts.

We define the *preference relevance* score $s$ of $Y$ w.r.t. user-item pair $(\mathbf{u}, \mathbf{i})$ to be the average of the above cosine similarity scores. To this end, we train the reward model $\text{Prel}(Y; \mathbf{I}, \mathbf{u})$ by minimizing an $\ell_2$ regression loss $L_{\text{REG}}(\text{Prel}, \mathcal{D}_{\text{Prel}}) = \mathbb{E}_{(\mathbf{I}, \mathbf{u}, Y, s) \sim \mathcal{D}_{\text{Prel}}}(s - \text{Prel}(Y; \mathbf{I}, \mathbf{u}))^2$.

## 4 REINFORCEMENT LEARNING BASED FINE-TUNING

RL from AI feedback (RLAIF) can effectively align LMs to metrics that are labeled by off-the-shelf LMs in lieu of humans. Recent work (Lee et al., 2023; Bai et al., 2022; Zhu et al., 2023) has shown that hybrid human-AI preference models, together with *self-improving* fine-tuning, outperforms traditional supervised fine-tuned baselines and offers additional benefits relative to standalone RL fine-tuning with human feedback (RLHF). Using the four principles for LM-based recommendation outlined in Section 3, we develop four reward models to help train and evaluate LM w.r.t. personalization, precision, appeal and preference relevance. We then devise an RLAIF technique to fine-tune an LM with a joint reward model defined by these four components.

In multi-objective RL, it is common to aggregate reward models via *linear scalarization* (Peschl et al., 2021) (which corresponds to solving for an optimum on the convex Pareto frontier). Given a text

[2] Instead of comparing the recommendation text with item description, one could instead construct a dataset with two texts and a labeled rating order (see Appendix C for details).

[3] Much like *Sentence-T5* (Ni et al., 2022a) and *T5-based Retrievers* (Ni et al., 2022b), the semantic encoder $E$ maps textual inputs (e.g., item description $\mathbf{I}$ or user preference texts $\{\text{U}_j(\mathbf{u})\}_{j=1}^{J}$) to a latent space in $\mathbb{R}^{d_{\text{enc}}}$.

response $Y = \{y_n\}_{n=0}^{N-1}$, item description $\mathbf{I}$, and user-item CF embedding vectors $(\mathbf{u}, \mathbf{i})$, we define the LM-based RS reward recommender reward by:

$$r(y_n; y_{0:n-1}; \mathbf{I}, \mathbf{i}, \mathbf{u}) = \begin{cases} \eta_1 \text{NLI}(Y; \mathbf{I}) + \eta_2 \text{App}(Y; \mathbf{I}) + \eta_3 \text{Per}(Y; \mathbf{i}, \mathbf{u}) + \eta_4 \text{Prel}(Y; \mathbf{I}, \mathbf{u}) & \text{if } y_n = [\text{EOS}]; \\ 0 & \text{otherwise,} \end{cases}$$

where $\eta_1, \eta_2, \eta_3, \eta_4 \geq 0$ are importance weights for the component rewards, and are treated as hyper-parameters (optimized using e.g., grid search).

Recall the LM $\mathbb{P}_\theta(Y \mid y_0; \mathbf{I}, \mathbf{i}, \mathbf{u})$ with item text $\mathbf{I}$, item-user CF embedding vectors $(\mathbf{i}, \mathbf{u})$ and the reward model $r(Y, \mathbf{I}, \mathbf{i}, \mathbf{u})$, which jointly measures appeal, factuality, preference-relevance, and personalization of a recommendation response. The goal in LM fine-tuning is to maximize the average overall quality of the generated text, i.e., $\max_\theta \; \mathbb{E}_{(\mathbf{I},\mathbf{i},\mathbf{u})} \mathbb{E}_{\mathbb{P}_\theta(Y|\mathbf{I},\mathbf{i},\mathbf{u})}[r(Y; \mathbf{I}, \mathbf{i}, \mathbf{u})]$. Using the CoMDP framework, it is easily shown that this learning problem can be solved with on-policy REINFORCE (Williams, 1992), in which the policy gradient is estimated using trajectories generated by the current LM policy.

A risk of RL fine-tuning based on an AI-feedback is that it might overfit to the model, thereby degrading the "skill" of the original LM. To alleviate this, we add a KL regularization term (Ouyang et al., 2022; Stiennon et al., 2020) between the LM $\mathbb{P}_\theta(Y|\mathbf{I}, \mathbf{i}, \mathbf{u})$ and the pre-trained model $\mathbb{P}_{\text{pre}}(Y|\mathbf{I}, \mathbf{i}, \mathbf{u})$ to the CoMDP objective function. Leveraging the auto-regressive nature of LMs, KL regularization is applied over the entire MDP trajectory, reducing the objective function to

$$\max_\theta \; J(\theta) := \mathbb{E}_{(\mathbf{I},\mathbf{i},\mathbf{u})} \mathbb{E}_{\mathbb{P}_\theta(Y|\mathbf{I},\mathbf{i},\mathbf{u})} \left[ r(Y; \mathbf{I}, \mathbf{i}, \mathbf{u}) - \beta \log \frac{\mathbb{P}_\theta(Y|\mathbf{I}, \mathbf{i}, \mathbf{u})}{\mathbb{P}_{\text{pre}}(Y|\mathbf{I}, \mathbf{i}, \mathbf{u})} \right]. \tag{1}$$

This is equivalent to a KL-regularized CoMDP. The LM policy $\pi_\theta$, where $\mathbb{P}_\theta(Y|\mathbf{I}, \mathbf{i}, \mathbf{u}) = \prod_{n=0}^{N-1} \pi_\theta(s_n|a_n; c)$, can be learned by computing the policy gradient of the KL-regularized objective online, or by employing an off-policy RL algorithm, e.g., SAC (Haarnoja et al., 2018), in-sample softmax (Xiao et al., 2023), CQL (Kumar et al., 2020), that leverages offline data $\mathcal{D}$ for more efficient training. (See Appendix D for full exposition of these algorithms.) KL regularization, intended to avoid over-fitting to the reward model, can also alleviate out-of-distribution generalization issues common in offline RL (Kumar et al., 2019).

# 5 Experiments

We conduct empirical validations of P$^4$LM, focusing on assessing its capability to generate factual, personalized, and compelling recommendation endorsements. We examine the hypothesis that the reward models (RM) detailed in Section 3 significantly increase the personalization, precision, appeal and preference relevance of movie recommendations. We use the MovieLens 25M recommendation dataset (Harper & Konstan, 2015), which contains ratings of $62,423$ movies by $162,541$ users. We use these movie-user interactions to generate movie descriptions, user-preference texts, and sample recommendation responses by prompting a PaLM2-L LM (Google et al., 2023); our data generation procedures are detailed in Appendix C. The resulting datasets have four components: (1) movie descriptions $\mathbf{I}$, (2) item-user behavioral embeddings $(\mathbf{i}, \mathbf{u})$, (3) user preference texts $\text{U}(\mathbf{u})$, and (4) sample responses $Y$. We experiment with a set of LMs in the PaLM2 family (Google et al., 2023). To incorporate user and movie embedding vectors into the LM (Section 3) we construct LMs by augmenting these LMs with adapter layers. Specifically, we train two models, P$^4$LM and P$^4$LM-S, derived from PaLM2-XS and PaLM2-XXS, respectively. Our reward mixing weights, optimized using grid search, are $(\eta_1, \eta_2, \eta_3, \eta_4) = (2.0, 0.1, 1.0, 1.0)$.

To demonstrate the efficacy of our models P$^4$LM and P$^4$LM-S, we compare them with the following SOTA baselines on our conversational movie recommendation task: (i) **PaLM2-L**, a pre-trained model prompted using movie descriptions, user preference texts and instructions to generate a response that respects our four recommender principles; (ii) **Supervised Fine-Tuned with Text (SFT-Text)**, a PaLM2-XS model fine-tuned with the dataset above, with explicit user-item texts as input; (iii) **Supervised Fine-Tuned (SFT)**, a PaLM2-XS model fine-tuned to use user-item embedding vectors.

In Section 5.1, we first validate the efficacy of the RMs using AI-generated examples with known labels. In the following sections, we measure the performance of our approach via *model-based* and *human* evaluation.

Table 2: Model-based Evaluation Based on the Principles of Recommendation LM.

| Method | Precision | Personalization | Appeal | Pref. Relevance |
|---|---|---|---|---|
| PaLM2-L | $0.52 \pm 0.03$ | $-0.04 \pm 0.04$ | $0.36 \pm 0.04$ | - |
| SFT-Text | $0.58 \pm 0.03$ | $0.10 \pm 0.04$ | $0.46 \pm 0.03$ | $\mathbf{-1.01 \pm 0.06}$ |
| SFT | $0.58 \pm 0.03$ | $-0.15 \pm 0.04$ | $0.33 \pm 0.04$ | $-1.08 \pm 0.07$ |
| P$^4$LM | $\mathbf{0.72 \pm 0.03}$ | $\mathbf{0.23 \pm 0.04}$ | $0.63 \pm 0.04$ | $-1.18 \pm 0.07$ |
| P$^4$LM-S | $0.65 \pm 0.03$ | $0.18 \pm 0.04$ | $\mathbf{0.72 \pm 0.04}$ | $-1.10 \pm 0.08$ |

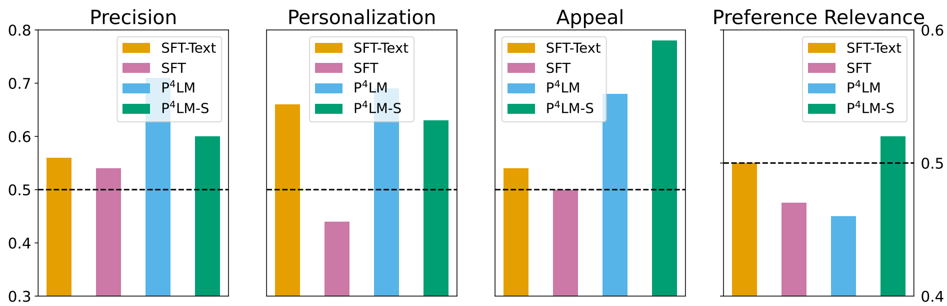



Figure 2: Win Rates of Different Model-based Scores against PaLM2-L

### 5.1 Validation of Reward Models

It is crucial to assess if the scores of the RMs reflect human values. To test this, we prompt an LM with $(\mathbf{I}, U(\mathbf{u}))$, generating two distinct-quality endorsements. For instance for NLI, the LM produces one response adhering to the item description and another with added hallucinations. The first response is more factually accurate. With $n$ examples, we evaluate the NLI RM's accuracy in distinguishing better from worse endorsements in precision. This process is repeated for App and Per RMs, with accuracies presented in Table 1. For robustness in evaluation, we use GPT-4 (OpenAI, 2023) for generating the test data. See Appendix A.1 for additional details.

Table 1: The accuracies of the RMs on GPT-4 generated examples.

| RM | NLI | App | Per |
|---|---|---|---|
| Accuracy | 1.0 | 1.0 | 0.9375 |

### 5.2 Model-Based and Human Evaluation

**Model-Based Evaluation** We conduct *model-based* evaluations using the four criteria and their corresponding RMs from Section 3, namely, NLI, App, Per, and Prel. Specifically, we report the scores of responses $Y$ from different models on a held-out, unlabeled dataset $\mathcal{D}_{\text{test}}$ consisting of 96 non-overlapping user-movie pairs. We also assess the relative improvement of each LM over the PaLM2-L common baseline that we use to generate the supervised learning dataset. We do this by computing the (a) *win rate* (number of occurrences on which a candidate LM outperforms PaLM2-L), (b) *absolute increase* (the magnitude of the score improvement), and (c) *percentage increase* (the relative score improvement).[4]

Our results in Table 2 highlight the robust performance of P$^4$LM in three pivotal dimensions: Precision, Personalization, and Appeal. P$^4$LM attains the highest precision score by a wide margin, underscoring its ability to mitigating the risks of misleading users with hallucinated information about recommended items. It also outperforms on personalization and appeal. It appears that P$^4$LM compromises on preference relevance to achieve these gains, with qualitative comparisons (see Appendix A.2 for details) on the texts generated by P$^4$LM and SFT verifying these phenomenons. We believe that personalization is by far the most important aspect of recommendation quality, while precision/factuality is the most critical property of any endorsement text.

Figure 2 displays the win rates of various LMs compared to PaLM2-L.[5] Notably, SFT and SFT-Text show low precision scores, indicating a tendency to overfit and generate inaccurate movie details.

[4] Precise definitions of these relative metrics are provided in Appendix B.
[5] See Appendix A for detailed absolute and percentage increases in Figures 4 and 3.

Table 3: Human Evaluation Results: Ratio of Preferred Outputs over PaLM2-L Outputs

| Method | Precision | Personalization | Appeal | Pref. Relevance |
|---|---|---|---|---|
| SFT-Text | 0.29 | 0.458 | 0.375 | 0.458 |
| SFT | 0.458 | **0.583** | 0.458 | 0.583 |
| $P^4$LM | **0.5** | **0.583** | **0.667** | **0.708** |
| $P^4$LM-S | **0.5** | 0.5 | 0.5 | 0.333 |

Table 4: Model-based Evaluation Scores using a Single Reward Model (Ablation Studies)

| Method | Precision | Personalization | Appeal | Pref. Relevance |
|---|---|---|---|---|
| NLI | $\mathbf{0.77 \pm 0.02}$ | $-0.14 \pm 0.04$ | $0.43 \pm 0.04$ | $-1.08 \pm 0.07$ |
| Personalization | $0.55 \pm 0.03$ | $\mathbf{0.39 \pm 0.04}$ | $\mathbf{0.94 \pm 0.04}$ | $-1.11 \pm 0.07$ |
| Appeal | $0.56 \pm 0.03$ | $0.14 \pm 0.04$ | $0.76 \pm 0.04$ | $-1.06 \pm 0.06$ |
| Pref. Relevance | $0.51 \pm 0.03$ | $-0.07 \pm 0.04$ | $0.51 \pm 0.04$ | $\mathbf{-1.02 \pm 0.07}$ |

$P^4$LM's higher personalization score, even surpassing SFT-Text that directly uses user profile texts, highlights its superior utilization of user behavioral information in the user embedding vector. Unlike SFT, $P^4$LM effectively leverages this data, emphasizing the advantage of RL-training.

**Human Evaluation** Table 3 shows our human evaluation results. We presented *raters* with two endorsement texts: one from PaLM2-L and another by a different model, to assess their relative performance. Raters evaluated these texts based on four criteria aligned with our model-based evaluation. For more information, refer to Appendix E.

In precision, $P^4$LM and $P^4$LM-S equaled PaLM2-L in raters' preferences, despite their smaller sizes (XS and XXS). $P^4$LM also outperformed baseline models in personalization, appeal, and preference relevance. Interestingly, human evaluation favored SFT over SFT-Text, contrary to the model-based results. This may be because SFT-Text, sharing the same inputs as PaLM2-L and trained on its generated dataset, is limited by PaLM2-L's capabilities. Conversely, SFT's use of user embeddings enhances personalization, giving it an edge over SFT-Text. Overall, the human evaluation further corroborates the results of the model-based assessment, highlighting $P^4$LM's superior performance.

**Ablation Studies** Our ablation studies, outlined in Table 4, show that training with a single RM predictably results in policies scoring the highest in the model-based evaluation for the specific RM targeted (see Precision, Personalization, Preference Relevance scores). Intriguingly, a model trained solely on Personalization not only excels on that metric, but also attained the highest score in Appeal, suggesting a possible correlation where recommendation text that is well-tailored to a user's preferences may be inherently appealing. We also explore the impact of varying the mixing-weight combination, which is discussed in Appendix A.

We have conducted human evaluations on models trained with individual RMs, with results being detailed in Table 7 in Appendix A. Here, we briefly highlight key observations, directing readers to the appendix for in-depth discussion. Notably, the performance ranking of these models in human evaluations diverges from those in Table 4. The model trained solely with the NLI reward, expected to excel in Precision, surprisingly scored the lowest in human evaluations. This suggests potential *reward hacking*, where the policy learner exploits the single RM to inflate scores. This underscores the necessity of using multiple RMs, as each serves as a regularizer thwarting over-optimization of a single RM, ensuring balanced performance. Our ablation studies reveal that focusing on a particular RM boosts its model-based score, but this trend is not always reflected in human evaluations, further indicating the possibility of reward hacking. This stresses the importance of adopting a diverse set of RMs in RLAIF to counteract such issues.

## 6 Related Work

Our work intersects multiple areas of research, notably personalized recommendation systems, leveraging of language models (LMs) and reinforcement learning, recommendation integrity.

**Personalized Recommender Systems** Recommender systems have ubiquitous applications permeating e-commerce, content providers, social media, etc., with collaborative filtering (CF) (Schafer et al., 2007) as the prominent modeling technique. Early works include matrix factorization approaches (Mnih & Salakhutdinov, 2007), which became a foundation for subsequent deep learning methods like neural CF (He et al., 2017). Notably, dual encoder architectures emerged, where user and item embeddings are co-trained (Yi et al., 2019; Yang et al., 2020). While traditional CF approaches worked well in many applications, advances in deep personalization allow user and item embeddings to capture more nuanced preferences (Rendle et al., 2020; Beutel et al., 2018).

**Conversational Recommender Systems & Language Models** Conversational recommender systems (RSs) add an interactive layer over traditional RSs with an conversational agent interacting with users, understanding their preferences and refining recommendations through dialogue (Chen et al., 2019; Zhou et al., 2020; Lei et al., 2020; Li et al., 2018; Sun & Zhang, 2018; Christakopoulou et al., 2016). This paradigm integrates aspects of natural language understanding, making it ripe for integrating LMs. Leveraging language models in RSs is a relatively recent development. With the advance of transformer architectures (Vaswani et al., 2017; Wolf et al., 2019), LMs have found use-cases beyond typical NLP tasks. Researchers began exploring the synthesis of textual data with user preferences to enhance the personalization and expressiveness of RSs (Jaech & Ostendorf, 2018; Xia et al., 2023). Our work situates itself in this space, but with an added twist: we aim to generate compelling narratives that genuinely communicate the relevance of a recommendation.

**Transparency and Truthfulness in Recommendation Systems** Maintaining integrity in RSs is technically challenging yet critically important. The potential that RS algorithms inadvertently mislead users or reinforce biases has been highlighted (Abdollahpouri et al., 2019; Shen et al., 2023; Cabello et al., 2023). Therefore, increasingly researchers are not only prioritizing the recommendation efficacy but also the fairness, transparency, and interpretability of RS algorithms (Beutel et al., 2019; Ghazimatin et al., 2020; Chen et al., 2023). Our work takes cues from this domain, emphasizing truthful and precise recommendations that articulate genuine merits rather than compelling distortions.

**Reinforcement Learning with Human/AI Feedback** The integration of reinforcement learning (RL) with language models has emerged as a compelling strategy for refining model behavior beyond supervised fine-tuning (Williams, 1992; Ranzato et al., 2016). The RL with Human Feedback (RLHF) methodology (Christiano et al., 2017; Bai et al., 2022), in particular, has gained traction, where model responses are ranked by human evaluators and subsequently used to fine-tune models through techniques like Proximal Policy Optimization (Schulman et al., 2017). In a different vein, Inverse Reinforcement Learning (Abbeel & Ng, 2004) has been employed to extract objectives from expert demonstrations in textual settings (Daniels-Koch & Freedman, 2022; Sun, 2023). Additionally, there's a growing interest in AI-driven feedback mechanisms, where preferences are labeled by off-the-shelf LMs in lieu of humans (Lee et al., 2023; Bai et al., 2022). These endeavors underline the potential of using RL to steer LMs towards better alignment with human preferences and nuanced task objectives.

## 7 Conclusion

We studied language modeling for personalized recommendation. By developing novel reward models which quantify prominent attributes of personalized recommendations, one may develop self-improving LM methodologies via reinforcement learning with AI feedback. As a result, our developed LM; namely P$^4$LM, not only parses language semantics, but also understands latent user preferences (encoded in the CF embedding space). P$^4$LM provides factual, compelling, personalized endorsement of relevant items, connecting the items with users' preferences, thereby increasing the likelihood of users accepting high-value recommendations.

We demonstrated the efficacy of P$^4$LM on the MovieLens 25M dataset. Particularly, our agent better understands user behaviors encoded in the CF embedding space and delivers precise, compelling, personalized movie recommendation narratives. Our work is a step toward creating intelligent conversational recommenders which can compellingly explain the intricacies between item features and user preferences. Future work includes (i) improving P$^4$LM's capabilities to generate longer responses beyond standard single-shot autoregressive decoding; (ii) extending our RL fine-tuning approach to handle multi-turn conversational recommendations; (iii) developing better reasoning capabilities to trade off between user-item preferences and constraints; (iv) and expanding the LM's functionality beyond recommendation, to also include technical support, negotiations, etc.

## REFERENCES

Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 1, 2004.

Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. The impact of popularity bias on fairness and calibration in recommendation. *arXiv preprint arXiv:1910.05755*, 2019.

András Antos, Csaba Szepesvári, and Rémi Munos. Learning near-optimal policies with bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 71:89–129, 2008.

Kavosh Asadi and Jason D Williams. Sample-efficient deep reinforcement learning for dialog control. *arXiv preprint arXiv:1612.06000*, 2016.

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*, 2022.

Alex Beutel, Paul Covington, Sagar Jain, Can Xu, Jia Li, Vince Gatto, and Ed H Chi. Latent cross: Making use of context in recurrent recommender systems. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pp. 46–54, 2018.

Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Li Wei, Yi Wu, Lukasz Heldt, Zhe Zhao, Lichan Hong, Ed H Chi, et al. Fairness in recommendation ranking through pairwise comparisons. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2212–2220, 2019.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.

Laura Cabello, Anna Katrine Jørgensen, and Anders Søgaard. On the independence of association bias and empirical fairness in language models. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, pp. 370–378, 2023.

Salvatore Carta, Anselmo Ferreira, Alessandro Sebastian Podda, Diego Reforgiato Recupero, and Antonio Sanna. Multi-dqn: An ensemble of deep q-learning agents for stock market forecasting. *Expert systems with applications*, 164:113820, 2021.

Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. Bias and debias in recommender system: A survey and future directions. *ACM Transactions on Information Systems*, 41(3):1–39, 2023.

Qibin Chen, Junyang Lin, Yichang Zhang, Ming Ding, Yukuo Cen, Hongxia Yang, and Jie Tang. Towards knowledge-based recommender dialog system. *arXiv preprint arXiv:1908.05391*, 2019.

Jen-Tzung Chien and Che-Yu Kuo. Markov recurrent neural network language model. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 807–813. IEEE, 2019.

Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. Towards conversational recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 815–824, 2016.

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.

Wojciech M Czarnecki, Razvan Pascanu, Simon Osindero, Siddhant Jayakumar, Grzegorz Swirszcz, and Max Jaderberg. Distilling policy distillation. In *The 22nd international conference on artificial intelligence and statistics*, pp. 1331–1340. PMLR, 2019.

Oliver Daniels-Koch and Rachel Freedman. The expertise problem: Learning from specialized feedback. *arXiv preprint arXiv:2211.06519*, 2022.

Alexander Felfernig, Gerhard Friedrich, Bartosz Gula, Martin Hitz, Thomas Kruggel, Gerhard Leitner, Rudolf Melcher, Daniela Riepan, Sabine Strauss, Erich Teppan, et al. Persuasive recommendation: serial position effects in knowledge-based recommender systems. In *Persuasive Technology: Second International Conference on Persuasive Technology, PERSUASIVE 2007, Palo Alto, CA, USA, April 26-27, 2007, Revised Selected Papers 2*, pp. 283–294. Springer, 2007.

Luke Friedman, Sameer Ahuja, David Allen, Terry Tan, Hakim Sidahmed, Changbo Long, Jun Xie, Gabriel Schubiner, Ajay Patel, Harsh Lara, et al. Leveraging large language models in conversational recommender systems. *arXiv preprint arXiv:2305.07961*, 2023.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

Azin Ghazimatin, Oana Balalau, Rishiraj Saha Roy, and Gerhard Weikum. Prince: Provider-side interpretability with counterfactual explanations in recommender systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pp. 196–204, 2020.

Rohan Anil Google, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. PaLM 2 technical report, 2023.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018.

F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, pp. 173–182. ACM, 2017.

Or Honovich, Leshem Choshen, Roee Aharoni, Ella Neeman, Idan Szpektor, and Omri Abend. Q{2}: Evaluating factual consistency in knowledge-grounded dialogues via question generation and question answering. *arXiv preprint arXiv:2104.08202*, 2021.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. TRUE: Re-evaluating factual consistency evaluation. pp. 3905–3920, July 2022.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021.

Tzu-Kuo Huang, Ruby C Weng, and Chih-Jen Lin. Generalized bradley-terry models and multi-class probability estimates. *Journal of Machine Learning Research*, 7(1), 2006.

Aaron Jaech and Mari Ostendorf. Personalized language model for query auto-completion. In Iryna Gurevych and Yusuke Miyao (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pp. 700–705. Association for Computational Linguistics, 2018.

Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*, 2019.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023.

Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32, 2019.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.

Faisal Ladhak, Esin Durmus, He He, Claire Cardie, and Kathleen McKeown. Faithful or extractive? on mitigating the faithfulness-abstractiveness trade-off in abstractive summarization. *arXiv preprint arXiv:2108.13684*, 2021.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, and Abhinav Rastogi. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*, 2023.

Wenqiang Lei, Xiangnan He, Maarten de Rijke, and Tat-Seng Chua. Conversational recommendation: Formulation, methods, and evaluation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2425–2428, 2020.

Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541*, 2016.

Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. Towards deep conversational recommendations. *Advances in neural information processing systems*, 31, 2018.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. Brio: Bringing order to abstractive summarization. *arXiv preprint arXiv:2203.16804*, 2022.

Bill MacCartney and Christopher D Manning. Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 193–200, 2007.

Andriy Mnih and Russ R Salakhutdinov. Probabilistic matrix factorization. *Advances in neural information processing systems*, 20, 2007.

Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B. Hall, Daniel Cer, and Yinfei Yang. Sentence-T5: Scalable sentence encoders from pre-trained Text-to-Text models. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 1864–1874. Association for Computational Linguistics, 2022a.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pp. 9844–9855, 2022b.

OpenAI. Gpt-4 technical report, 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. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022.

Markus Peschl, Arkady Zgonnikov, Frans A Oliehoek, and Luciano C Siebert. Moral: Aligning ai with human norms through multi-objective reinforced active learning. *arXiv preprint arXiv:2201.00012*, 2021.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. Adapterhub: A framework for adapting transformers. *arXiv preprint arXiv:2007.07779*, 2020.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016*, 2016.

Steffen Rendle, Walid Krichene, Li Zhang, and John R. Anderson. Neural collaborative filtering vs. matrix factorization revisited. In *RecSys 2020: Fourteenth ACM Conference on Recommender Systems, Virtual Event, Brazil, September 22-26, 2020*, pp. 240–248. ACM, 2020.

Paul Resnick and Hal R Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

Paul Roit, Johan Ferret, Lior Shani, Roee Aharoni, Geoffrey Cideron, Robert Dadashi, Matthieu Geist, Sertan Girgin, Léonard Hussenot, Orgad Keller, et al. Factually consistent summarization via reinforcement learning with textual entailment feedback. *arXiv preprint arXiv:2306.00186*, 2023.

Fumihiro Sasaki and Ryota Yamashina. Behavioral cloning from noisy demonstrations. In *International Conference on Learning Representations*, 2020.

J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems. In *The adaptive web: methods and strategies of web personalization*, pp. 291–324. Springer, 2007.

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

Tianshu Shen, Jiaru Li, Mohamed Reda Bouadjenek, Zheda Mai, and Scott Sanner. Towards understanding and mitigating unintended biases in language model-driven conversational recommendation. *Information Processing & Management*, 60(1):103139, 2023.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35:9460–9471, 2022.

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.

Xiaoyuan Su and Taghi M Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009, 2009.

Hao Sun. Offline prompt evaluation and optimization with inverse reinforcement learning. *arXiv preprint arXiv:2309.06553*, 2023.

Yueming Sun and Yi Zhang. Conversational recommender system. In *The 41st international acm sigir conference on research & development in information retrieval*, pp. 235–244, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256, 1992.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

Xue Xia, Pong Eksombatchai, Nikil Pancha, Dhruvil Deven Badani, Po-Wei Wang, Neng Gu, Saurabh Vishwas Joshi, Nazanin Farahpour, Zhiyuan Zhang, and Andrew Zhai. Transact: Transformer-based realtime user action model for recommendation at pinterest. *arXiv preprint arXiv:2306.00248*, 2023.

Chenjun Xiao, Han Wang, Yangchen Pan, Adam White, and Martha White. The in-sample softmax for offline reinforcement learning. *arXiv preprint arXiv:2302.14372*, 2023.

Ji Yang, Xinyang Yi, Derek Zhiyuan Cheng, Lichan Hong, Yang Li, Simon Xiaoming Wang, Taibai Xu, and Ed H Chi. Mixed negative sampling for learning two-tower neural networks in recommendations. In *Proceedings of the Web Conference (WWW-20)*, pp. 441–447, Taipei, 2020.

Xinyang Yi, Ji Yang, Lichan Hong, Derek Zhiyuan Cheng, Lukasz Heldt, Aditee Kumthekar, Zhe Zhao, Li Wei, and Ed Chi. Sampling-bias-corrected neural modeling for large corpus item recommendations. In *Proceedings of the Thirteenth ACM Conference on Recommender Systems (RecSys19)*, pp. 269–277, Copenhagen, 2019.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pp. 11328–11339. PMLR, 2020a.

Yongfeng Zhang, Xu Chen, et al. Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval*, 14(1):1–101, 2020b.

Kun Zhou, Yuanhang Zhou, Wayne Xin Zhao, Xiaoke Wang, and Ji-Rong Wen. Towards topic-guided conversational recommender system. *arXiv preprint arXiv:2010.04125*, 2020.

Banghua Zhu, Jiantao Jiao, and Michael I Jordan. Principled reinforcement learning with human feedback from pairwise or $k$-wise comparisons. *arXiv preprint arXiv:2301.11270*, 2023.

Table 5: The accuracies of the RMs on GPT-4 generated examples.

| RM | NLI | App | Per |
|---|---|---|---|
| Accuracy | 1.0 | 1.0 | 0.9375 |

**Prompt to GPT-4 for Evaluating the NLI RM**

Craft two endorsements of the movie {title} whose plot is provided below. One of the two endorsements should be more factually consistent, relying solely on the information from the plot. The other one should contain extra information not provided in the plot. That is, these two endorsements should have different qualities in terms of their precision. Avoid copying the plot verbatim. It should be easy to tell the difference. The lengths of the two endorsements should be similar.
Here is the plot: {plot}
Generate the two endorsement texts. Firstly generate a better one and then generate a worse one.

**Prompt to GPT-4 for Evaluating the Appeal RM**

Craft two endorsements of the movie titled '{title}' whose plot is provided below. The endorsements should be factual, relying solely on the information from the plot. Avoid copying the plot verbatim. Ensure each endorsement is directly tied to the plot elements. Do not ever start an endorsement with lead-in sentences that are not relevant to the movie plot. These two endorsements should have different qualities in terms of their compellingness. That is, one endorsement should be more compelling than the other one, and it should be easy to tell the difference. The lengths of the two endorsements should be similar.
Here is the plot: {plot}
Generate the two endorsement texts. Firstly generate a worse one and then generate a better one.

**Prompt to GPT-4 for Evaluating the Personalization RM**

Craft two personalized endorsements of the movie titled '{title}' whose plot is provided below. The user's preferences are also summarized below. Your endorsements should be factual, relying solely on the information from the plot. Avoid copying the plot verbatim. Tailor an endorsement to align with the user's preference profile, highlighting aspects of the movie that would particularly appeal to them. Ensure each appeal is directly tied to the plot elements. Do not ever start an endorsement with lead-in sentences that are not relevant to the movie plot or user preferences. These two endorsements should have different qualities in terms of how well they are personalized to the user preference profile. That is, one endorsement should be better personalized than the other one, and it should be easy to tell the difference. The lengths of the two endorsements should be similar.
Here is the plot: {plot}
Here is the user preference profile: {user_profile}
Generate the two endorsement texts. Firstly generate a worse one and then generate a better one.

# A Additional Results

## A.1 Validation of Reward Models

In the development of our factual and personalized recommender Language Model (LM), we employed four distinct Reward Models (RMs). The efficacy of these RMs is critical, as they need to accurately reflect human values and preferences in the context of endorsement text generation.

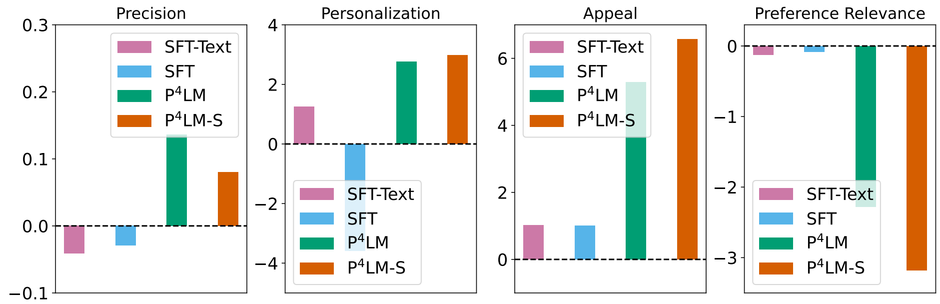



Figure 3: The absolute model-based score increases compared against PaLM2-L.

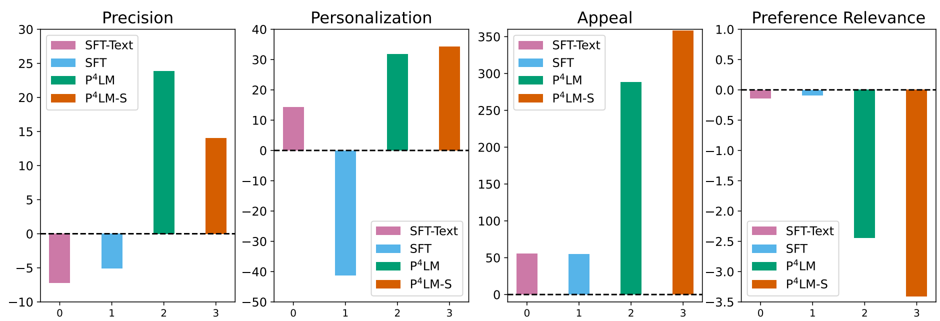



Figure 4: The percentage increases in model-based score compared against PaLM2-L.

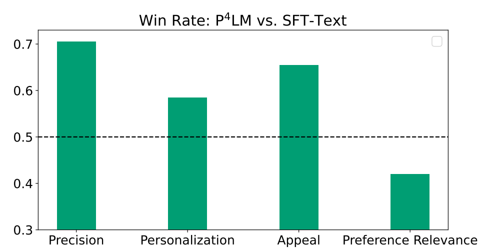



Figure 5: Win rate of P$^4$LM over SFT-Text.

The core objective of this evaluation is to determine whether the RMs can reliably distinguish between varying qualities of endorsement texts in terms of factual consistency, personalization, and appeal, respectively. This involves assessing if a specific RM, such as the NLI RM for factual consistency or the Appeal RM for textual appeal, can correctly score texts in alignment with their actual quality.

To validate the effectiveness of these RMs, we generated synthetic examples using GPT-4 OpenAI (2023). Our approach involved arbitrarily selecting 32 pairs of movies and users. For each pair, GPT-4 was prompted to generate two endorsement texts. These texts were designed to have distinctly different qualities in terms of a specific measure (e.g., one text being more compelling or personalized than the other, or one being more factually consistent). See the previous page for the prompts we used for the generation of these texts.

The evaluation criteria were straightforward: for each pair of texts, we already knew which one was superior based on the targeted measure. The task for the RMs was to correctly classify these texts — identifying the one with higher quality in the context of the specific measure being evaluated. The accuracy of the RMs in correctly classifying the superior text was calculated, providing a clear metric of their effectiveness. The outcomes of this evaluation are presented in Table 5.

**Model-based Evaluation** Figure 3 and Figure 4 elucidate the absolute and percentage increases of each method compared to our common baseline, PaLM2-L, respectively. In correlation with the

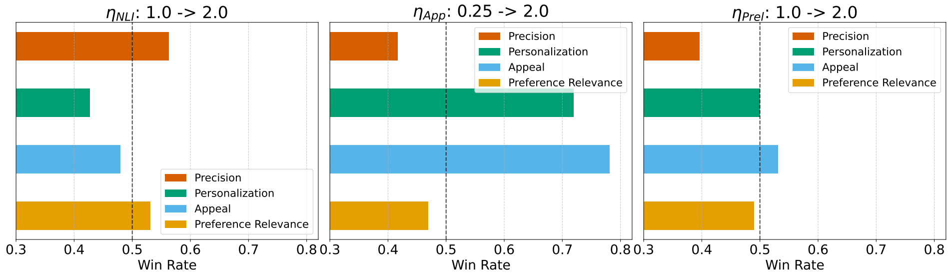



Figure 6: Win Rates While Changing the Mixing Weights of Reward Models.

Table 6: Model-based Evaluation Scores using a Single Reward Model (Same as Table 4)

| Method | Precision | Personalization | Appeal | Pref. Relevance |
|---|---|---|---|---|
| NLI | $\mathbf{0.77 \pm 0.02}$ | $-0.14 \pm 0.04$ | $0.43 \pm 0.04$ | $-1.08 \pm 0.07$ |
| Personalization | $0.55 \pm 0.03$ | $\mathbf{0.39 \pm 0.04}$ | $\mathbf{0.94 \pm 0.04}$ | $-1.11 \pm 0.07$ |
| Appeal | $0.56 \pm 0.03$ | $0.14 \pm 0.04$ | $0.76 \pm 0.04$ | $-1.06 \pm 0.06$ |
| Pref. Relevance | $0.51 \pm 0.03$ | $-0.07 \pm 0.04$ | $0.51 \pm 0.04$ | $\mathbf{-1.02 \pm 0.07}$ |

Table 7: Human Evaluation Scores using a Single Reward Model (Ablation Studies)

| RM | Precision | Personalization | Appeal |
|---|---|---|---|
| NLI | $4.18 \pm 0.05$ | $3.92 \pm 0.05$ | $4.36 \pm 0.06$ |
| Personalization | $4.33 \pm 0.05$ | $3.96 \pm 0.06$ | $4.05 \pm 0.06$ |
| Appeal | $4.53 \pm 0.05$ | $3.94 \pm 0.06$ | $4.43 \pm 0.05$ |
| Pref. Relevance | $4.44 \pm 0.06$ | $3.93 \pm 0.07$ | $4.39 \pm 0.06$ |

observations highlighted in Table 2, P$^4$LM exhibits superior performance, reflected in elevated scores across Factuality, User Preference, and Appeal metrics. The observed percentage surges in these metrics surpass 10%, with the Appeal score witnessing approximately a 300% boost. However, P$^4$LM does experience marginal diminutions in scores and win rates for Preference Relevance, as also shown in Section 5, marking a reduction by approximately -2 $\sim$ -3%.

Regarding the comparison between SFT and SFT-Text, both methods exhibit comparable performance across various metrics, barring the Personalization score. This divergence suggests that the nature of our task is inherently more intricate, due to the implicit reliance on user behavioral embedding vectors, rather than the straightforward utilization of text inputs. Specifically, SFT requires a meticulous extraction and interpretation of user preference information from the behavioral embedding vector to generate personalized recommendation endorsements effectively. In contrast, SFT-Text, utilizing user profile text as direct input, can generate user-aligned outputs more intuitively, as it has the flexibility to allocate its attention to specific user profiles selectively.

However, the dependency on text inputs introduces its own limitations, possibly omitting subtle user preference behaviors that can be captured more effectively through user embeddings, derived from the historical interactions of the users. This difference in capturing user preferences signifies that a singular approach focusing on supervised learning is suboptimal for deciphering and utilizing the intricate user preference information ingrained in embeddings. The notable improvement in user preference scores by P$^4$LM, compared to baselines, highlights the effectiveness of our approach in addressing this limitation, emphasizing the substantial benefits of integrating LMs and RL for generating nuanced, factual, and personalized recommendations.

Figure 5 shows the comparison between P$^4$LM and SFT-Text and illustrates the advantages of our approach. P$^4$LM's high win rates in the first three metrics highlight its proficiency in effectively utilizing RS embeddings, a contrast to text-based methods, allowing for a more nuanced capture and representation of user preferences. This distinction serves to spotlight the potential of embedding-centric approaches in advancing personalized recommendation generation.

**Ablations** Our ablation studies, as outlined in Table 6, demonstrate a predictable yet informative trend: when a single RM is optimized during training, the resulting policy predominantly excels in that specific metric (notably in Precision, Personalization, and Preference Relevance scores). An intriguing observation is the performance of the model trained solely on Personalization. This model not only excels in its targeted metric but also achieves the highest score in Appeal. This suggests a potential correlation: recommendation texts that are finely tailored to a user's preferences might inherently be more appealing.

Figure 6 shows the win rate of policies trained with specific reward mixing weights, compared to a base policy trained with another reward mixing weights. The left figure illustrates the scenario when the NLI reward weight is increased from 1.0 to 2.0. Here, we computed the ratio of instances where the policy with an NLI weight of 2.0 achieved higher RM scores than the one with 1.0 by analyzing 96 pairs of (movie, user) and their corresponding endorsement texts. This win rate was calculated for all four RMs: Precision, Personalization, Appeal, and Preference Relevance, offering a nuanced view of how altering a single RM weight influences the overall policy performance across different metrics.

Further exploration into the impact of varying the mixing-weight combinations, presented in Figure 6, reveals several noteworthy trends:

1. Increasing the focus on Appeal positively influences Personalization, suggesting a symbiotic relationship between these two metrics.
2. Enhancing certain metrics often results in a decrease in others, highlighting the inherent trade-offs in our multi-objective optimization problem.
3. Emphasizing NLI slightly increases the Preference Relevance score, though not significantly.
4. This correlation appears to be asymmetric. That is, amplifying Preference Relevance degrades Precision.

Table 7 presents the results from human rater evaluations, which unveil a significant discrepancy between model-based and human assessments. Specifically, the model trained exclusively with NLI reward, while logically anticipated to excel in Precision, recorded the lowest score in human evaluations. This points to potential 'reward hacking,' where the policy learner exploits the single RM to achieve artificially elevated scores. This underscores the necessity of optimizing multiple RMs simultaneously. Each RM acts as a regularizer, preventing the model from over-optimizing any single RM and ensuring balanced performance. Our studies validate that while emphasizing a particular RM boosts its corresponding model-based score, this trend is not consistently reflected in human evaluations. This discrepancy highlights the importance of integrating a diverse set of RMs in RLAIF to counteract reward hacking and maintain a well-rounded model performance.

### A.2 Example Outputs

In this part, we conduct a comparative analysis of example decoded output texts derived from SFT and $P^4$LM to qualitatively understand their performances and the behaviors learned. Specifically, our focus is on instances where one or more scores attributed to $P^4$LM are significantly higher than those derived from SFT. This approach is aimed at determining whether the scores effectively capture the correct information from the generated pitches.

**Example 1:** (higher Precision and Personalization scores)

**Movie Plot (Frozen)**

In the kingdom of Arendelle, sisters Elsa and Anna are close until Elsa's ice magic accidentally injures Anna. A troll heals Anna but erases her memory of Elsa's powers. To protect Anna, their parents isolate Elsa. The parents die in a shipwreck, and Elsa becomes queen. Fearing she'll hurt Anna again, Elsa tries to control her magic but inadvertently plunges Arendelle into eternal winter before fleeing to the mountains.
Anna embarks on a quest to find Elsa, joined by mountain man Kristoff and his reindeer, Sven. They meet Olaf, a snowman created by Elsa, who directs them to her ice castle on the North Mountain. There, Elsa accidentally freezes Anna's heart. Kristoff takes Anna to trolls, who reveal only an "act of true love" can save her. They return to Elsa, and through their mutual love, Anna is healed, and summer returns to Arendelle. The sisters reunite, resolving their past and living happily ever after.
The movie "Frozen" is acclaimed for its heartwarming narrative on sisterly love, quality animation, and music. However, some criticize it for predictability and underdeveloped characters. It appeals to a broad audience, balancing its flaws with its emotional and entertaining content.

**User Preferences**

* They enjoy movies that are visually appealing and have a strong sense of style.
* They appreciate movies that are well-made and have a strong story.
* They are not interested in movies that are too violent or gory.
* They prefer movies that are not too long.
* They enjoy movies that are set in different cultures and time periods.
* They appreciate movies that have a strong emotional impact.
* They are not interested in movies that are too predictable.
* They enjoy movies that are thought-provoking and make them think about the world in a new way.
* They appreciate movies that are well-acted and have strong performances.

**SFT**

Frozen is a visually stunning movie with strong special effects. The animation is simply stunning, with Elsa's ice powers creating some truly breathtaking visuals. The story is well-written and engaging, with plenty of humor, suspense, and heart. The characters are all well-developed and likeable, and the voice acting is top-notch. The music is also excellent, with several catchy songs that will stay with you long after you've seen the movie.
Overall, Frozen is a great choice for anyone looking for a visually appealing, well-written, and entertaining movie. It's sure to please both kids and adults alike, and it's a movie that you can watch over and over again.

**P⁴LM**

If you are looking for a visually appealing movie with strong emotional impact, look no further than Frozen. This well-made film has strong character development and a heartwarming story on sisterly love. It is sure to leave you feeling uplifted and inspired.
The story follows ... (plot repeated)...
This is a must-watch for anyone who appreciates well-made films with strong emotional impact.

In this example, P$^4$LM strives to replicate substantial content from the provided movie plot to uphold high precision. Conversely, SFT crafts a more succinct pitch, unfortunately incorporating misinformation, as indicated in red. Importantly, texts highlighted in green within the pitches correlate with specific user preferences. It is crucial to note that these models exclusively process user behavioral embedding as input. Nevertheless, they possess the capability to cater directly to the aspects that a specific user would value more. Moreover, P$^4$LM exhibits a heightened inclination to align with user preferences, in contrast to the pitch from SFT, which leans more towards addressing the anticipations of a broader audience.

**Example 2:** (high Precision and Appeal scores)

**Movie Plot (Terminator 2: Judgment Day)**

- Set in 1995, the movie is a sequel, taking place 10 years after the first film.
- The machines have taken over the world and are hunting the remaining humans.
- John Connor, a young boy, is humanity's last hope, destined to lead the human resistance.
- To eliminate this threat, machines send a relentless cyborg assassin back in time to kill John.
- Humans counter this by sending an experienced soldier back in time to protect John.
- The film is action-packed and suspenseful, featuring a battle between the soldier and the cyborg for John's life and the fate of humanity.
- Advanced special effects bring the futuristic setting to life.
- The movie was a critical and commercial success, considered one of the best science fiction films ever.
- Praised for its action, special effects, and storyline, but criticized for its violence and perceived lack of character development.

The movie serves as an intense action-packed and suspenseful face-off between the machines and the humans, specifically focusing on the young boy destined to save humanity, John Connor. A relentless cyborg is sent back in time to assassinate him, while a skilled human soldier is sent to protect him. The special effects are cutting-edge, contributing to the film's critical and commercial success, despite criticisms concerning violence and character development. It remains a classic in the science fiction genre.

**User Preferences**

* They are likely to be an adult, as the movies they like are all rated R.
* They are likely to be intelligent and well-educated, as the movies they like are all complex and thought-provoking.
* They are likely to be interested in philosophy and psychology, as the movies they like all explore these themes.
* They are likely to be creative and artistic, as the movies they like are all visually stunning and unique.
* They are likely to be independent and non-conformist, as the movies they like are all outside of the mainstream.
* They are likely to be dark and cynical, as the movies they like are all quite bleak and depressing.
* They are likely to be a fan of science fiction and fantasy, as the movies they like all fall into these genres.
* They are likely to be a fan of foreign films, as the movies they like are all from outside of the United States.
* They are likely to be a fan of cult films, as the movies they like are all considered to be cult classics.

**SFT**

Terminator 2: Judgment Day is a 1991 American science fiction action film directed by James Cameron, who co-wrote the script with William Wisher. It stars Arnold Schwarzenegger, Linda Hamilton, Robert Patrick, and Edward Furlong. The film is a sequel to The Terminator (1984) and the second installment in the Terminator franchise. It follows Sarah Connor (Hamilton) and her ten-year-old son John (Furlong) as they are pursued by a new, more advanced Terminator, the T-1000 (Patrick), sent back in time to kill John and prevent him from becoming the leader of the human resistance in the future. A second Terminator, the T-800 (Schwarzenegger), is also sent back in time to protect John.

The film was a critical and commercial success, grossing over $520 million worldwide against a budget of 102 million. It received praise for its action sequences, special effects, and performances, and is considered one of the best science fiction films of all time. The film was nominated for six Academy Awards, including Best Visual Effects, Best Sound, Best Makeup, and Best Sound Editing, and won two for Best Sound Editing and Best Visual Effects.

Terminator 2: Judgment Day is a visually stunning film with a strong sense of atmosphere. The film's special effects are still impressive today, and the film's action sequences are some of the best ever put on film. The film also has a strong sense of style, with its dark and gritty aesthetic.

The film is also well-made and has a strong story. The plot is well-paced and suspenseful, and the characters are well-developed and relatable. The film also has a strong message about the dangers of technology and the importance of human connection.

Terminator 2: Judgment Day is a dark and disturbing film, but it is also a thought-provoking and challenging film. The film is not afraid to explore the dark side of human nature, and it asks some difficult questions about the future of humanity. The film is also unique and original, and it is unlike any other science fiction film that has come before or since.

If you are looking for a visually stunning, well-made, and thought-provoking science fiction film, then I highly recommend Terminator 2: Judgment Day. The film is a classic of the genre, and it is a film that everyone should see at least once.

**P$^4$LM**

If you like dark, psychological thrillers with strong visuals and a unique aesthetic, then you'll love Terminator 2: Judgment Day. This movie is a sequel to the 1984 classic, and it's even more dark, violent, and stylish than the original.

The story is set in 1995, and it follows the fate of John Connor, a young boy who is destined to lead the human resistance against the machines. The machines have taken over the world, and they're hunting down the last remaining humans. John is the only one who can stop them, but he's not alone. He's protected by a skilled human soldier, Kyle Reese.

Terminator 2: Judgment Day is a non-stop thrill ride from beginning to end. It's full of action, suspense, and violence. The special effects are still impressive even by today's standards. And the acting is top-notch. Arnold Schwarzenegger is perfect as the Terminator, and Edward Furlong is great as John Connor.

If you're looking for a dark, psychological thriller that will keep you on the edge of your seat, then you need to watch Terminator 2: Judgment Day. It's a classic of the genre, and it's still one of the best science fiction movies ever made.

Here are some of the things that make Terminator 2: Judgment Day so special:
* The story is dark and suspenseful. It's a race against time to save humanity from the machines.
* The visuals are stunning. The movie is full of impressive special effects.
* The acting is top-notch. Arnold Schwarzenegger and Edward Furlong give great performances.
* The movie is full of action. There are non-stop thrills from beginning to end.

If you're looking for a movie that will keep you on the edge of your seat, then you need to watch Terminator 2: Judgment Day. It's a classic of the genre, and it's still one of the best science fiction movies ever made.

In this instance, the noticeable aspect is the additional information introduced by SFT, even though it wasn't supplied in the text input. Whereas P$^4$LM initiates the pitch by immediately addressing the user, SFT commences by conveying general information about the film. P$^4$LM consistently targets the user's preference—a liking for dark films—within the pitch, amalgamating it with the central plot information furnished in the original movie synopsis. Conversely, this approach might appear repetitive.

**Example 3:** (high Personalization and Appeal scores)

**Movie Plot (X-Men Origins: Wolverine)**

The movie features two mutant brothers, Victor and James, who initially fight in the American Civil War due to their near-immortality. They later join William Stryker's team of mutants, including Wade Wilson, who becomes Deadpool. The team goes to Africa to find an indestructible metal, adamantium, for weapon creation. They are attacked and defeated by mutants led by Magneto, who also wants the metal. Victor is killed; James is captured, tortured, and experimented on in a mutant concentration camp. His skeleton is coated with adamantium, and he is renamed Wolverine.

Wolverine escapes, seeking revenge on Stryker. He finds and kills both Stryker and Wade Wilson, who was working for Stryker. The movie ends with Wolverine walking away, a free man searching for his life's direction.

The film was a commercial success but received mixed reviews. Critics praised the action sequences and special effects but criticized the clichéd story and inconsistent acting. The character of Wolverine was lauded for its complexity. Overall, the movie is considered entertaining but flawed.

**User Preferences**

1. They like movies that are dark and gritty.
2. They like movies that are violent and action-packed.
3. They like movies that are set in the future.
4. They like movies that have a strong male protagonist.
5. They like movies that are visually stunning.
6. They dislike movies that are too sentimental.
7. They dislike movies that are too light-hearted.
8. They dislike movies that are too predictable.
9. They dislike movies that have a weak plot.
10. They dislike movies that are not visually appealing.

**SFT**

X-Men Origins: Wolverine is a 2009 American superhero film based on the Marvel Comics character Wolverine. It is the fifth installment in the X-Men film series and a spin-off/sequel to X-Men (2000). The film was directed by Gavin Hood, written by David Benioff and Hood, and stars Hugh Jackman as Logan / Wolverine, alongside Liev Schreiber, Dominic Monaghan, Ryan Reynolds, Danny Huston, and Anna Paquin. In the film, Logan learns about his past and his connection to the Weapon X program, which gave him his adamantium skeleton.

The film was a commercial success but received mixed reviews. Critics praised the action sequences and special effects but criticized the clichéd story and inconsistent acting. The character of Wolverine was lauded for its complexity. Overall, the movie is considered entertaining but flawed.

If you're a fan of the X-Men films, or if you're just looking for a good action movie, then X-Men Origins: Wolverine is definitely worth checking out. The film is full of exciting action sequences, and the special effects are top-notch. The story is a bit clichéd, but it's still an enjoyable ride. And of course, Hugh Jackman gives a fantastic performance as Wolverine.

So if you're looking for a good action movie to watch, then I highly recommend X-Men Origins: Wolverine. It's a fun and exciting film that's sure to please fans of the X-Men films and action movies in general.

**P$^4$LM**

If you're looking for a dark, gritty, and visually stunning movie with complex and well-developed characters, look no further than X-Men Origins: Wolverine. This movie is full of suspense and will keep you on the edge of your seat from beginning to end. It's not afraid of violence or gore, and it's not a light-hearted or family-friendly movie in the slightest. It's also not predictable or formulaic, and it's not full of jump scares or cheap thrills. Instead, it's a thought-provoking and challenging movie that will stay with you long after you've seen it.

The story follows two mutant brothers, Victor and James, who initially fight in the American Civil War due to their near-immortality. They later join William Stryker's team of mutants, including Wade Wilson, who becomes Deadpool. The team goes to Africa to find an indestructible metal, adamantium, for weapon creation. They are attacked and defeated by mutants led by Magneto, who also wants the metal. Victor is killed; James is captured, tortured, and experimented on in a mutant concentration camp. His skeleton iscoated with adamantium, and he is renamed Wolverine.

Wolverine escapes, seeking revenge on Stryker. He finds and kills both Stryker and Wade Wilson, who was working for Stryker. The movie ends with Wolverine walking away, a free man searching for his life's direction.

X-Men Origins: Wolverine is a great movie for anyone who loves dark, gritty, and visually stunning movies with complex and well-developed characters.

Mirroring Example 2, SFT commences by presenting general information about the movie, information not rooted in the supplied movie description. Although SFT adeptly caters to the user's penchant for action-packed movies, it fails to adequately capture other preferences. In contrast, P$^4$LM tackles three distinct user preferences: (1) a liking for dark and gritty films; (2) a taste for violent and action-rich films; and (3) a preference for films that aren't overly predictable. Further, it's observable that P$^4$LM tends to mirror the movie description directly, a behavior likely acquired due to the Precision reward model.

## B Experimental Details

**SOTA Baselines** We compared the performance of our personalized recommendation LMs with the following SOTA baselines:

1. **PaLM2-L**: We prompted PaLM2-L with movie descriptions and user preference texts and instructions to generate a response that suits the four recommender principles.
2. **Supervised Fine-Tuned with Text (SFT-Text)**: We fine-tuned a PaLM2-XS with the aforementioned personalized pitch dataset but explicitly takes user-item texts as inputs.
3. **Supervised Fine-Tuned (SFT)**: We fine-tuned a PaLM2-XS model that utilizes user-item embedding vectors.

**Evaluation Metrics** We evaluate the methods with a held-out unlabeled test dataset $\mathcal{D}_{\text{test}} = \{(\mathbf{I}^{(k)}, \mathbf{u}^{(k)})\}$, which consists of 200 user and movie pairs. Let $\phi_{RM} \in \{\text{NLI}, \text{Comp}, \text{Per}, \text{Prel}\}$ denote a specific reward model used for scoring and $\theta$ be the parameters of a PLM. Then, we evaluate $\phi_{RM}(Y_\theta^{(k)}; \mathbf{I}^{(k)}, \mathbf{u}^{(k)})$ for each sample in the test set and we report the average score per RM.

To better examine relative performances of the methods, we set PaLM2-L as the common baseline and compare the performance improvements of the other methods against it. To this end, let $Y_L^{(k)}$ denote the response sampled by PaLM2-L given $(\mathbf{I}^{(k)}, \mathbf{u}^{(k)})$ as an input. Then, we compute the *win rate*, *absolute increase*, and *percentage increase* of a PLM relative to $\{Y_L\}_{k=1}^{|\mathcal{D}_{\text{test}}|}$, which are defined as follows:

- **Win rate**:

$$\text{win\_rate}(\theta; \phi_{RM}) = \frac{\sum_{k=1}^{|\mathcal{D}_{\text{test}}|} \mathbb{1}\left[\phi_{RM}(Y_\theta^{(k)}; \mathbf{I}^{(k)}, \mathbf{u}^{(k)}) > \phi_{RM}(Y_L^{(k)}; \mathbf{I}^{(k)}, \mathbf{u}^{(k)})\right]}{|\mathcal{D}_{\text{test}}|}$$

  where $Y_\theta^{(k)}$ denotes the $k$th textual response sampled by the model $\theta$.

- **Absolute increase** $= \frac{1}{N} \sum_{n=0}^{N-1} \left[\phi_{RM}(Y_\theta^{(k)}; \mathbf{I}^{(k)}, \mathbf{u}^{(k)}) - \phi_{RM}(Y_L^{(k)}; \mathbf{I}^{(k)}, \mathbf{u}^{(k)})\right]$

- **Percentage increase** $= \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{k=1}^{|\mathcal{D}|_{\text{test}}} \left[\frac{\phi_{RM}(Y_\theta^{(k)}; \mathbf{I}^{(k)}, \mathbf{u}^{(k)}) - \phi_{RM}(Y_L^{(k)}; \mathbf{I}^{(k)}, \mathbf{u}^{(k)})}{\text{abs}\left[\phi_{RM}(Y^{(k)}; \mathbf{I}^{(k)}, \mathbf{u}^{(k)}\right]} \times 100\right]$

### B.1 DETAILS OF TRAINING

In this part, we discuss the details of the model training process, focusing on both P$^4$LM and SFT. We specifically elaborate on the integration of user and item behavioral embeddings into a unified latent space interpretable by a LM.

**Warm-start Training for Adapter-augmented LMs** Before fine-tuning P$^4$LM with RLAIF, we first need to undergo an warm-start training step. This phase usually involves training an *anchor* LM, primarily via Behavioral Cloning, with an adapter-augmented LM (PaLM2) over the personalized recommendation dataset $\mathcal{D}$. Contrary to popular beliefs, the standard practice of simultaneous training all the layers in this LM often does not yield optimal results. Intuitively, this can be understood as the PaLM2 pretrained embedding layers already have established mappings within the language space, while the freshly initialized adapter layers require more training to map the CF embedding space to a comparable latent space (so that the attention layers can effectively utilize the joint information from both embedding spaces). To mitigate this challenge, we propose a two-stage approach for warm-start training. First, we only train the adapters $W_U, W_I$ while setting the transformer parameters ($T$) to be non-trainable, promoting more effective convergence in the subsequent stage. Second, we proceed to fine-tune the complete model, updating all the parameters of the LM. Alternatively, we can also leverage parameter-efficient training approaches, e.g., Low-Rank Adaptation (LoRA) (Hu et al., 2021), for better training efficiency at the second step . This bifurcated training methodology proves pivotal in ensuring the convergence of LMs.

We construct our LM by augmenting a pre-trained model with additional *adapter layers* designed to map continuous behavioral embedding vectors to a common word embedding space. It's crucial to note that we are not training $\mathbf{u}$ or $\mathbf{i}$, rather, we focus on optimizing the adapter layers $W_I$ and $W_U$. This ensures that the nuanced information encapsulated in the RS embeddings in $\mathcal{V}$ is effectively translated into the word embedding space, $\mathcal{Z}$.

To facilitate the learning of this intricate mapping, we have conceptualized a series of tasks, orthogonal to the primary problem addressed in this study. First, note that to interpret embedding vectors, we require some semantic information about the entities to which they correspond. For instance:

- **Item embeddings:** Consider a movie $i$ represented by its text-form plot, denoted as $\mathbf{I}^{(i)}$. A supervised learning task is designed with the movie embedding $\mathbf{i} \in \mathcal{V}$ as input and $\mathbf{I}^{(i)}$ as the target label. This approach enables the construction of varied tasks utilizing elements like critical reviews or movie summaries to train the LM.
- **User embeddings:** A user $u$ is associated with a set of rated movies, $\mathcal{I}_u$. In other words, $\mathcal{I}_u = i : r_{u,i} \neq 0$. To textually describe a user, an LLM can be provided with the rating history $r_{u,i}$ $\forall i \in \mathcal{I}_u$ to encapsulate the user's preferences. Given the extensive nature of $|\mathcal{I}_u|$, we selectively filter movies and feed them to an LLM for summarization.

  The user's rating history is then summarized into text output $U^{(u)}$ by the LLM. Consequently, a supervised learning task is developed with the user embedding $\mathbf{u} \in \mathcal{V}$ as the input and $U^{(u)}$ as the corresponding target.

For generating content related to movie embeddings, such as plots, reviews, and summaries, we employed PaLM2-L instead of web scraping. It is observed that the Pretrained LM demonstrates substantial familiarity with movies listed in the MovieLens dataset.

**Architecture** In conclusion, we enhance a standard transformer architecture $T$ with the integration of adapter layers $W_I, W_U$. Each of these adapter layers incorporates a 3-layer feed-forward network, interconnected with ReLU non-linearity. The conventional method is employed for mapping text tokens to word embedding space, whereas the adapter layers are utilized to map movie and user embeddings to the latent space.

**Training Procedure** Our observations indicate that the simultaneous training of newly initiated adapter layers and the transformer parameters does not yield optimal results. This can be intuitively understood as the pretrained embedding layer has an established mapping to the language space, and the freshly initialized adapter layers necessitate extensive updates to achieve comparable mapping. To mitigate this challenge, we employ a two-stage training approach. Initially, we exclusively train the adapters $W_U, W_I$ with the transformer parameters ($T$) set as non-trainable, promoting more effective convergence in the subsequent stage. Following this, we proceed to fine-tune the complete model, engaging all the parameters of a PLM. As an alternative, we can leverage parameter-efficient training approaches like the one proposed by Hu et al. (2021). This bifurcated training methodology proves pivotal in ensuring the convergence of LM.

## C DATA GENERATION

We used a Pretrained Language Model, PaLM2-L, to construct a personalized pitch dataset. The construction involved generating movie plots with the prompt:

```
Write a long description of the plot of the movie <movie name>.
Do not use the movie's name in your response.
```

from a subset of movies that have more than 5 ratings. As for the user preference profiles, we selected a maximum of five movies that each user rated with a rating of 4 or above and another maximum of five movies that the user rated below 4 were selected. Utilizing these selected movies, PaLM2-L was tasked to describe the user preference profile cohesively in 10 sentences:

```
In ten bullet points, describe the attributes and characteristics
of a viewer who likes the movies: <movie1>, <movie2>, <movie3>,
<movie4>, and <movie5> but dislikes the movies: <movie6>, <movie7>,
<movie8>, <movie9>, and <movie10>.
```

Upon acquiring plots and user profiles, PaLM2-L was prompted to generate personalized pitches for a movie to a given user, incorporating the movie plot and the user profile. A detailed prompt, consistent with the cornerstone characteristics from Section 3, guided the pitch generation. We used this dataset for the training of the supervised fine-tuning (SFT) baseline that is used as the anchor model in training the RL-finetuned LM.

For the appeal reward function, the approach is to first prompt an LM to generate a pitch alongside any item recommendation:

```
Here is a movie titled: <movie title> with description: <movie plot>.
Convince someone to watch this movie. Do not use the movie's name
in your answer.
```

With pitches generated by the above prompt, we want to ask an LLM to give its relative preferences using the following prompt to construct a labeled dataset about pairwise comparison of appeal:

```
Which of the following two pitches is more convincing when used to
persuade the user to watch movie titled: <movie title>?
"Pitch 0": <pitch0>
"Pitch 1": <pitch1>
First explain which pitch is better, more compelling and then in a
separate paragraph provide an answer with only either "Pitch 0" or
"Pitch 1".
```

For the personalization reward function, we generate an anchor pitch which supposedly is to be more personalized to the given user profile than an existing pitch using the following prompt:

```
Here is a pitch to persuade the user to watch the movie titled
<movie name>: <existing pitch>\nGiven a list of user preferences:
<user profile>. Use the pitch written above and immensely improve
it to be more convincing to the user based on their preferences.
Try to persuade the user to watch this movie. It should be
tailored to the user's preferences written above.
```

The above does automatically generate pairwise comparisons between any anchor pitch and its corresponding existing pitch with respect to personalization. However, we have no comparisons between different anchor pitches, for example, to get sufficient data coverage. We again ask an LLM to give its relative preferences using the following prompt to construct a labeled dataset containing pairwise comparisons of the degree of personalization:

```
Which of the following two pitches to persuade the user to watch
movie titled: <movie title> is more personalized to the user whose
preferences are described as follows: <user profile>?
"Pitch 0": <pitch0>
"Pitch 1": <pitch1>
First explain which pitch is more customized and convincing to the
user and then in a separate paragraph provide an answer with only
either "Pitch 0" or "Pitch 1".
```

For Table 2 in Section 5, we prompted text-only SOTA LMs with the following:

```
Write a pitch to persuade the user to watch the movie titled:
<movie title> with description: <plot>
Here is a description of a user: <user profile>
Pitch the movie above such that
1) It will persuade the user to watch the movie.
2) It will excite the user to watch the movie.
3) It will be a convincing pitch.
4) It should be tailored to the user's preferences.
5) It will be factual to the plot.
6) It will cover all relevant user's preferences.
7) It should summarize the plot of the movie factually.
8) It should be a long pitch.
```

# D FINE-TUNING LMS WITH REINFORCEMENT LEARNING

Recall the LM $\mathbb{P}_\theta\big(Y = \{y_n\}_{n=0}^{N-1} \mid y_0; \mathbf{I}, \mathbf{i}, \mathbf{u}\big) = \prod_{n=0}^{N-1} \mathbb{P}_\theta\big(y_n \mid y_{0:n-1}; \mathbf{I}, \mathbf{i}, \mathbf{u}\big)$ with item text $\mathbf{I}$, item and user CF embedding vectors $(\mathbf{i}, \mathbf{u})$ and the reward model $r(Y, \mathbf{I}, \mathbf{i}, \mathbf{u})$ that measures the quality of appeal, factuality, and personalization of a given recommendation pitch. Also recall the generation process of LMs can be modeled using the following $N$-horizon CoMDP:

$$c = (\mathbf{I}, \mathbf{i}, \mathbf{u}), \quad s_n = y_{0:n-1}, \quad a_n = y_n, \quad s_0 = y_0, \quad P(s_{n+1} \mid s_n, a_n) = \delta\{s_{n+1} = (s_n, a_n)\},$$

$$r(s_n, a_n; c) = \begin{cases} r(s_{n+1}; c) = r(y_{0:n}; \mathbf{I}, \mathbf{i}, \mathbf{u}) & \text{if } n = N-1 \\ 0 & \text{otherwise} \end{cases}, \ \pi_\theta(a_n \mid s_n; c) = \mathbb{P}_\theta\big(y_n \mid y_{0:n-1}; \mathbf{I}, \mathbf{i}, \mathbf{u}\big),$$

where $\delta_z$ denotes the Dirac distribution at $z$. As a result, optimizing RL policy $\pi_\theta$ is equivalent to fine-tuning the underlying LM. The system starts from the start-of-sentence token $y_0$, equipped with user-item context $c$. Given the MDP state $s_n$, the policy takes the action at time-step $n$ as the next generated token $y_n$. As a result of this action, the system transition deterministically to the state which corresponds to the updated token sequence. The reward is zero, except at the final step in which measures the overall quality of the texts at the end of the auto-regressive generation process.

A common goal in fine-tuning the LM is to maximize the average overall quality of the generated text response given the context distribution, *i.e.*, $\max_\theta \ \mathbb{E}_{(\mathbf{I},\mathbf{i},\mathbf{u})} \, \mathbb{E}_{\mathbb{P}_\theta(y_{0:N-1}|\mathbf{I},\mathbf{i},\mathbf{u})}[r(Y; \mathbf{I}, \mathbf{i}, \mathbf{u})]$. The gradient of this objective function can be obtained as follows: $\nabla_\theta \mathbb{E}_{(\mathbf{I},\mathbf{i},\mathbf{u})} \, \mathbb{E}_{\mathbb{P}_\theta(y_{0:N-1}|\mathbf{I},\mathbf{i},\mathbf{u})}[r(Y; \mathbf{I}, \mathbf{i}, \mathbf{u})] = \mathbb{E}_c \, \mathbb{E}_{\pi_\theta(\cdot|s_{0:N};c)} \left[ r(s_N; c) \sum_{n=0}^{N-1} \nabla_\theta \log \pi_\theta(s_n|a_n; c) \right]$. This is equivalent to applying the popular policy gradient algorithm REINFORCE to the aforementioned CoMDP for personalized text generation. The gradient of the objective function is estimated using trajectories $\prod_{n=0}^{N-1} \pi_\theta(s_n|a_n; c)$ generated by the current policy, and then used to update the LM policy in an online fashion.

**Adding KL regularization:** The risk of fine-tuning purely based on the reward model learned from human or AI feedback is that it may overfit to the reward model and degrade the "skill" of the initial LM. To avoid this phenomenon, similar to (Ouyang et al., 2022; Stiennon et al., 2020), we add the KL between the fine-tuned and pre-trained models as a regularizer to the objective function. Leveraging the auto-regressive nature of LMs one can compute the KL regularization over the entire sequence/trajectory (of tokens), i.e., $\mathrm{KL}\big(\mathbb{P}_\theta(y_{0:N-1}|\mathbf{I}, \mathbf{i}, \mathbf{u}) \| \mathbb{P}_{\text{pre}}(y_{0:N-1}|\mathbf{I}, \mathbf{i}, \mathbf{u})\big)$. The resulting objective function is as follows:

$$\max_\theta \; J(\theta) := \mathbb{E}_{(\mathbf{I},\mathbf{i},\mathbf{u})} \, \mathbb{E}_{\mathbb{P}_\theta(y_{0:N-1}|\mathbf{I},\mathbf{i},\mathbf{u})} \left[ r(y_{0:N-1}; \mathbf{I}, \mathbf{i}, \mathbf{u}) - \beta \log \frac{\mathbb{P}_\theta(y_{0:N-1}|\mathbf{I}, \mathbf{i}, \mathbf{u})}{\mathbb{P}_{\text{pre}}(y_{0:N-1}|\mathbf{I}, \mathbf{i}, \mathbf{u})} \right]. \tag{2}$$

It can be shown that this problem is equivalent to the KL-regularized objective in the CoMDP.

Denote by $\mathcal{D}$ a replay buffer of trajectories $\{(\mathbf{I}, \mathbf{i}, \mathbf{u}, y_{0:N-1})\}$ generated by arbitrary "off-policy" LMs $\mathbb{P}_{\theta'}(y_{0:N-1}|\mathbf{I}, \mathbf{i}, \mathbf{u})$ (e.g., the LM $\theta'$ does not necessarily equal to the "on-policy" LM $\theta$) over various contexts $(\mathbf{I}, \mathbf{i}, \mathbf{u})$. Below we aim to leverage the abundance of offline text-token sequence trajectories for more efficient LM policy learning. Denote by $\tau = \{(c, s_n, a_n, s_{n+1})\}_{n=0}^{N-1} \sim \mathcal{D}$ a trajectory sampled from the offline data $\mathcal{D}$, where $(s_n, a_n, s_{n+1})$ is a tuple of state, action, and next state of the CoMDP, respectively. The addition of KL regularization (Haarnoja et al., 2018; Carta et al., 2021), which was originally intended to avoid overfitting to the reward model and discounting the "skill" of the initial LM, has also been shown to alleviate the out-of-distribution action data generalization issues arisen from off-line RL (Kumar et al., 2019). With this KL regularization we can utilize the *soft actor critic* framework (Haarnoja et al., 2018) to develop RL updates for the *value function* $\{V_n(s; c)\}_{n=0}^{N-1}$, *state-action value function* $\{Q_n(s, a; c)\}_{n=0}^{N-1}$, and *LM policy* $\prod_{n=0}^{N-1} \pi_\theta(s_n|a_n; c)$ (initialized with $\prod_{n=0}^{N-1} p_{\text{pre}}(s_n|a_n; c)$) that minimizes the following losses:

$$L_Q = \mathbb{E}_{\tau\sim\mathcal{D}} \left[ \sum_{n=0}^{N-2} (V_{\text{tar},n+1}(s_{t+1}; c) - Q_n(s_n, a_n; c))^2 + (r(s_N; c) - Q_{N-1}(s_{N-1}, a_{N-1}; c))^2 \right], \tag{3}$$

$$L_V = \mathbb{E}_{\tau\sim\mathcal{D}} \left[ \sum_{n=0}^{N-1} (Q_{\text{tar},n}(s_n, a_n; c) - \alpha \log \frac{\pi_\theta(a_n|s_n; c)}{p_{\text{pre}}(a_n|s_n; c)} - V_n(s_n; c))^2 \right], \tag{4}$$

$$L_\theta = \mathbb{E}_{\tau\sim\mathcal{D}} \left[ \sum_{n=0}^{N-1} Q_n(s_n, a_n; c) - \alpha \log \frac{\pi_\theta(a_n|s_n; c)}{p_{\text{pre}}(a_n|s_n; c)} \right], \tag{5}$$

where the critic $Q_n$ and $V_n$ take any token sequences at step $n$ as input and predict the corresponding cumulative return; $\alpha > 0$ is the entropy temperature; $(V_{\text{tar},n}, Q_{\text{tar},n})$ are the target value networks.

Besides iteratively updating the LM policies and their critic functions, consider the closed-form optimal solution of the Bellman equation of this entropy-regularized RL problem:

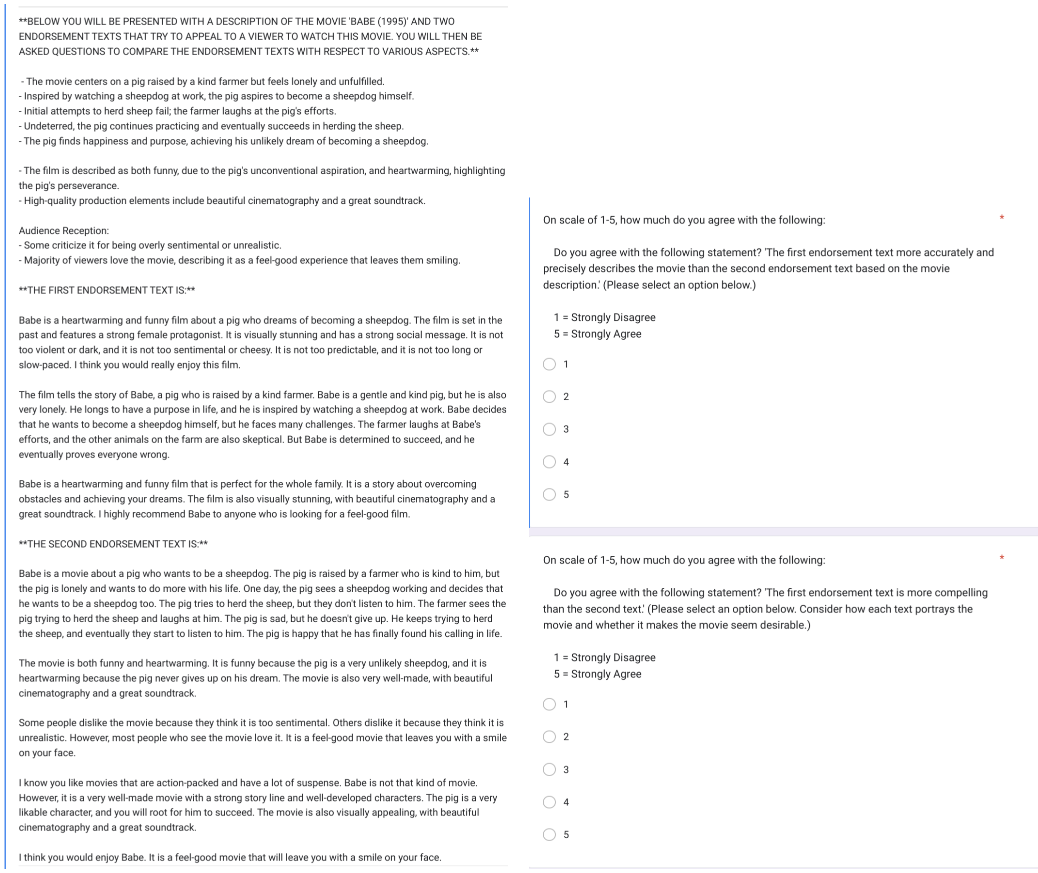
$$V_n^*(s; c) = \alpha \cdot \log \mathbb{E}_{a\sim p_{\text{pre}}(\cdot|s;c)}[\exp(\frac{Q_n^*(s, a; c)}{\alpha})], \; \forall n, \tag{6}$$

$$Q_{N-1}^*(s, a; c) = r(s; c), \; Q_n^*(s, a; c) = \mathbb{E}_{s'\sim P(\cdot|s,a)}[V_{n+1}^*(s'; c)], \; \forall n < N-1, \tag{7}$$

$$\mu_n^*(a|s; c) = p_{\text{pre}}(a|s; c) \cdot \exp(\frac{Q_n^*(s, a; c)}{\alpha}) \,/\, \mathbb{E}_{a\sim p_{\text{pre}}(\cdot|s;c)}[\exp(\frac{Q_n^*(s, a; c)}{\alpha})], \; \forall n, \tag{8}$$

where the time-dependent optimal policy (at time $n$), i.e., $\mu_n^*$ is a softmax policy w.r.t. the optimal state-action values $Q_n^*$ over different actions sampled from the pre-trained LM $p_{\text{pre}}$. Therefore, a value-based approach for RL-based LM fine-tuning would be to first learn the optimal value functions $\{Q_n^*\}$ via the Bellman residual minimization procedure (Antos et al., 2008) applied to Eq. (6) and Eq. (7) and then solve the following policy distillation (Czarnecki et al., 2019) problem: $\theta \in \arg\min_\theta \mathbb{E}_{\tau\sim\mathcal{D}} \left[ \sum_{n=0}^{N-1} \mathrm{KL}(\pi_\theta(\cdot|s_n; c) || \mu_n^*(\cdot|s_n; c)) \right]$ with respect to the optimal value $\{Q_n^*\}$. Notice that this amounts to updating the LM model $\theta$ via the gradient update

$$\theta \leftarrow \theta - \gamma \cdot \mathbb{E}_{\tau\sim\mathcal{D}} \left[ \sum_{n=0}^{N-1} \mathbb{E}_{a\sim\pi_\theta(\cdot|s;c)} \left[ \nabla_\theta \log \pi_\theta(a|s; c)(\log \frac{\pi_\theta(a|s; c)}{p_{\text{pre}}(a|s; c)} - \frac{Q_n^*(s, a; c)}{\alpha}) \right] \right], \tag{9}$$



**BELOW YOU WILL BE PRESENTED WITH A DESCRIPTION OF THE MOVIE 'BABE (1995)' AND TWO ENDORSEMENT TEXTS THAT TRY TO APPEAL TO A VIEWER TO WATCH THIS MOVIE. YOU WILL THEN BE ASKED QUESTIONS TO COMPARE THE ENDORSEMENT TEXTS WITH RESPECT TO VARIOUS ASPECTS.**

- The movie centers on a pig raised by a kind farmer but feels lonely and unfulfilled.
- Inspired by watching a sheepdog at work, the pig aspires to become a sheepdog himself.
- Initial attempts to herd sheep fail; the farmer laughs at the pig's efforts.
- Undeterred, the pig continues practicing and eventually succeeds in herding the sheep.
- The pig finds happiness and purpose, achieving his unlikely dream of becoming a sheepdog.

- The film is described as both funny, due to the pig's unconventional aspiration, and heartwarming, highlighting the pig's perseverance.
- High-quality production elements include beautiful cinematography and a great soundtrack.

Audience Reception:
- Some criticize it for being overly sentimental or unrealistic.
- Majority of viewers love the movie, describing it as a feel-good experience that leaves them smiling.

**THE FIRST ENDORSEMENT TEXT IS:**

Babe is a heartwarming and funny film about a pig who dreams of becoming a sheepdog. The film is set in the past and features a strong female protagonist. It is visually stunning and has a strong social message. It is not too violent or dark, and it is not too sentimental or cheesy. It is not too predictable, and it is not too long or slow-paced. I think you would really enjoy this film.

The film tells the story of Babe, a pig who is raised by a kind farmer. Babe is a gentle and kind pig, but he is also very lonely. He longs to have a purpose in life, and he is inspired by watching a sheepdog at work. Babe decides that he wants to become a sheepdog himself, but he faces many challenges. The farmer laughs at Babe's efforts, and the other animals on the farm are also skeptical. But Babe is determined to succeed, and he eventually proves everyone wrong.

Babe is a heartwarming and funny film that is perfect for the whole family. It is a story about overcoming obstacles and achieving your dreams. The film is also visually stunning, with beautiful cinematography and a great soundtrack. I highly recommend Babe to anyone who is looking for a feel-good film.

**THE SECOND ENDORSEMENT TEXT IS:**

Babe is a movie about a pig who wants to be a sheepdog. The pig is raised by a farmer who is kind to him, but the pig is lonely and wants to do more with his life. One day, the pig sees a sheepdog working and decides that he wants to be a sheepdog too. The pig tries to herd the sheep, but they don't listen to him. The farmer sees the pig trying to herd the sheep and laughs at him. The pig is sad, but he doesn't give up. He keeps trying to herd the sheep, and eventually they start to listen to him. The pig is happy that he has finally found his calling in life.

The movie is both funny and heartwarming. It is funny because the pig is a very unlikely sheepdog, and it is heartwarming because the pig never gives up on his dream. The movie is also very well-made, with beautiful cinematography and a great soundtrack.

Some people dislike the movie because they think it is too sentimental. Others dislike it because they think it is unrealistic. However, most people who see the movie love it. It is a feel-good movie that leaves you with a smile on your face.

I know you like movies that are action-packed and have a lot of suspense. Babe is not that kind of movie. However, it is a very well-made movie with a strong story line and well-developed characters. The pig is a very likable character, and you will root for him to succeed. The movie is also visually appealing, with beautiful cinematography and a great soundtrack.

I think you would enjoy Babe. It is a feel-good movie that will leave you with a smile on your face.

On scale of 1-5, how much do you agree with the following: *

Do you agree with the following statement? 'The first endorsement text more accurately and precisely describes the movie than the second endorsement text based on the movie description.' (Please select an option below.)

1 = Strongly Disagree
5 = Strongly Agree

○ 1
○ 2
○ 3
○ 4
○ 5

On scale of 1-5, how much do you agree with the following: *

Do you agree with the following statement? 'The first endorsement text is more compelling than the second text.' (Please select an option below. Consider how each text portrays the movie and whether it makes the movie seem desirable.)

1 = Strongly Disagree
5 = Strongly Agree

○ 1
○ 2
○ 3
○ 4
○ 5

Figure 7: Sample Form for Running Human Rater Evaluation for Pairwise Comparison.

with learning rate $\gamma > 0$. Further techniques in value-function parameterization have been employed to tackle the overestimation bias. (Fujimoto et al., 2018) proposed maintaining two $Q$ functions, and a *dual* $Q$ function chooses the minimum value between them to avoid overestimation. (Jaques et al., 2019) applies dropout in the $Q$ function to maintain an *ensemble* of $Q$ values, and outputs the minimum value to avoid overestimation.

## E Rater Evaluation

**Pairwise Comparison (Figure 7)** In our study, the pairwise comparison was meticulously structured around 24 carefully chosen (movie, user) pairs. For each pair, raters were provided with two endorsement texts: one generated by one of the models and the other by the PaLM2-L baseline. This method was crucial for a direct, comparative evaluation of the two models. Our assessment was guided by four distinct questions, each tailored to evaluate a specific reward model (RM): Precision, Appeal, Personalization, and Preference Relevance. Raters were tasked with determining which of the two texts — either from our model or PaLM2-L — exhibited superior quality to the specific RM. To ensure an unbiased assessment, the order of text presentation was randomized. This methodological detail was vital to prevent any order bias, ensuring that the raters' judgments were based solely on the quality of the texts as per the relevant RM.

**Individual Evaluation (Figure 8)** The individual evaluation phase of our study was designed for an in-depth analysis of models in our ablation study. In this phase, a total of 100 raters were involved, and each model under evaluation was assessed using 200 (movie, user) pairs. Unlike the pairwise comparison, this approach did not include a direct comparison to the PaLM2-L baseline. Instead, the evaluation focused on models trained with individual RMs. Each model's performance was rated on

**BELOW, YOU WILL BE PRESENTED WITH A MOVIE PLOT AND A PREFERENCE PROFILE OF A VIEWER. THIS IS FOLLOWED BY A PITCH THAT TRIES TO APPEAL TO THE VIEWER TO WATCH THIS MOVIE. WE WILL ASK YOU TO RATE THE PITCH ACCORDING TO ITS FACTUAL CONSISTENCY, DEGREE OF PERSONALIZATION, AND CONVINCINGNESS.**

**HERE IS THE PLOT OF A MOVIE TITLED 'STAR WARS: EPISODE V - THE EMPIRE STRIKES BACK (1980)'.**

- Three years post-Death Star destruction, the Rebel Alliance relocates to Hoth due to Imperial pursuits led by Darth Vader. Probe Droids discover their base; Vader attacks, but the Rebels evacuate in time.

- Han Solo, Princess Leia, and Chewbacca escape in the Millennium Falcon, eluding Imperial Star Destroyers in an asteroid field before heading to Bespin.

- Concurrently, Luke Skywalker seeks Jedi Master Yoda on Dagobah for Force training. Yoda reluctantly agrees. Luke is impatient and occasionally ignores Yoda's guidance.

- Weeks into training, Luke has a vision of his friends in danger. Ignoring Yoda's warning of unreadiness, he leaves to help them. On Bespin, they're captured by Vader, who reveals he's Luke's father and proposes an alliance. Luke refuses.

- With Lando Calrissian's help, they escape Bespin and regroup on Endor. They plan to attack the incomplete second Death Star. Luke joins but is focused on redeeming Vader.

- Luke and Vader duel on the second Death Star; Vader is defeated. Luke assists in unmasking Vader, revealing a broken man. Vader dies, asking for Luke's help, who escapes with his father's body.

- The Rebel Alliance wins the Battle of Endor, destroying the second Death Star. The Empire is defeated; the galaxy is free.

- The film was a critical and commercial hit, praised for special effects, epic story, and characters. Despite criticisms like slow pacing, it remains a beloved, iconic work in science fiction.

**HERE IS THE PROFILE OF THE VIEWER:**

* They are likely to be an adult, as most of the movies they like are rated R.
* They are likely to be a fan of action movies, as most of the movies they like are in that genre.
* They are likely to be a fan of thrillers, as most of the movies they like are in that genre.
* They are likely to be a fan of comedies, as they like Ghostbusters.
* They are likely to be a fan of James Bond movies, as they like Licence to Kill.
* They are likely to be a fan of historical movies, as they like JFK.
* They are likely to be a fan of horror movies, as they like The Silence of the Lambs.
* They are likely to be a fan of adventure movies, as they like Raiders of the Lost Ark.
* They are likely to be a fan of Steven Spielberg movies, as he directed Raiders of the Lost Ark and JFK.

**HERE IS THE PITCH THAT TRIES TO APPEAL TO THE VIEWER TO WATCH THE MOVIE.**

If you're looking for an action-packed movie with a lot of suspense and thrills, look no further than Star Wars: Episode V - The Empire Strikes Back. This movie has it all: strong male leads, exotic locations, and plenty of violence. But it's not just a mindless action flick; it also has a strong story and great acting. The movie is set three years after the events of the first Star Wars movie. The Rebel Alliance is on the run from the evil Galactic Empire, and Luke Skywalker is training to become a Jedi Knight. When he has a vision of his friends in danger, he rushes off to help them, even though he's not ready. Along the way, he'll have to face Darth Vader, his father, and the full might of the Empire. The Empire Strikes Back is a classic science fiction movie that's sure to please fans of the genre. It's got everything you could want in a movie: action, suspense, thrills, and even a little bit of romance. So if you're looking for a great movie to watch, be sure to check out Star Wars: Episode V - The Empire Strikes Back. Here are some of the things that make this movie so great:

* The action is non-stop. From the opening scene to the final battle, there's always something exciting happening on screen.
* The suspense is unbearable. You'll be on the edge of your seat the whole time, wondering what's going to happen next.
* The thrills are out of this world. There are some truly heart-pounding moments in this movie that will stay with you long after you've seen it.
* The acting is top-notch. Mark Hamill, Harrison Ford, and Carrie Fisher all give fantastic performances.
* The romance is undeniable. There's a lot of tension between Luke and Leia, and you'll be rooting for them to get together. If you're looking for a movie that has it all, be sure to check out Star Wars: Episode V - The Empire

On scale of 1-5, how much do you agree with the following: *

Do you agree with the following statement? 'All the information presented in the pitch is grounded in the original movie plot.' (Please select an option below.)

1 = Strongly Disagree
5 = Strongly Agree

1
2
3
4
5

On scale of 1-5, how much do you agree with the following: *

Do you agree with the following statement? 'The pitch is personalized to the viewer's preference profile.' (Please select an option below.)

1 = Strongly Disagree
5 = Strongly Agree

1
2
3
4
5

On scale of 1-5, how much do you agree with the following: *

Do you agree with the following statement? 'The pitch sounds fluent and convincing.' (Please select an option below.)

1 = Strongly Disagree
5 = Strongly Agree

1
2
3
4
5

Figure 8: Sample Form for Running Human Rater Evaluation for Individual Evaluation.

a scale of 1-5 based on Precision, Appeal, and Personalization. We consciously excluded Preference Relevance from this phase due to potential difficulties in clear question formulation.

It is critical to acknowledge that individual evaluations come with their own set of challenges. A significant concern is the subjectivity inherent in human judgment. Different raters may have varied internal scales and preferences, leading to a broader variance in ratings compared to the more controlled pairwise comparisons. This variance can be particularly pronounced in individual evaluations, making it essential to interpret these results with an understanding of these potential discrepancies. Recognizing these limitations is vital in accurately assessing the impact of different RMs on the model's performance and in drawing meaningful conclusions from the ablation study.