

Marketing Science Institute Working Paper Series 2026

Report No. 26-134

Generative Personalization with Textual and Non-textual Data

Mingzhang Yin, Ruijiang Gao and Ziwei Cong

"Generative Personalization with Textual and Non-textual Data" © 2026

Mingzhang Yin, Ruijiang Gao and Ziwei Cong

MSI Working Papers are Distributed for the benefit of MSI corporate and academic members and the general public. Reports are not to be reproduced or published in any form or by any means, electronic or mechanical, without written permission.

Generative Personalization with Textual and Non-textual Data

Mingzhang Yin[*] Ruijiang Gao[†] Ziwei Cong[‡]

Abstract

The widespread adoption of large language models (LLMs) on major platforms has transformed how firms create and deliver content to consumers. It allows platforms to create content rather than merely select from existing alternatives. However, existing LLMs often overlook heterogeneous consumer preferences or personalize at a high cost. We propose Generative Personalized Targeting with Generative Pre-trained Transformers ($\texttt{GPT}^2$), a framework that jointly models users' non-textual interactions and their preferred textual content. An attention-based user model extracts multiple latent preference factors from each user's interaction history and transforms them into soft prompts, which guide LLM fine-tuning and steer generation toward individual tastes. We apply $\texttt{GPT}^2$ for personalized headline generation using consumption data from Microsoft News and for question answering on Zhihu, a large knowledge-sharing platform. Across both, $\texttt{GPT}^2$ outperforms generic and personalized LLM benchmarks, producing content closer to lab-based self-written references and preferred by LLM-based evaluators. Interpretive analyses show that the model captures a wide range of heterogeneous linguistic preferences, which are mirrored in the generated text. Finally, we show how platforms can complement traditional discriminative targeting with generative targeting, and how generative targeting provides a solution to the cold-start problem for content with little prior feedback.

Keywords: Large Language Models, Personalization, User Modeling, Generative Targeting, Online Platforms

[*]Mingzhang Yin is an Assistant Professor of Marketing, University of Florida. Email: mingzhang.yin@warrington.ufl.edu
[†]Ruijiang Gao is an Assistant Professor of Information Systems at UT Dallas. Email: ruijiang.gao@utdallas.edu
[‡]Ziwei Cong is an Assistant Professor of Marketing, Georgetown University. Email: zc260@georgetown.edu

1 Introduction

In recent years, generative artificial intelligence (GenAI) has been widely adopted across major platforms as a new interface to provide content and interact with consumers (see Figure 1). Due to GenAI's high efficiency and scalability, a large study suggests that 74.2% of 900K newly created web pages in April 2025 contain AI-generated content (Law et al., 2025), and 90% of online content created in the late 2020s will be assisted by AI (Constantino, 2024). This shift fundamentally expands the content strategies available to online platforms. Rather than merely selecting from existing alternatives, platforms can now create content on demand, from generating answers for Q&A platforms and search engines to summarizing creative content and consumer reviews. A key question raised by this shift is whether generated content meets consumers' diverse preferences (Kapoor and Kumar, 2025).

Existing GenAI systems, exemplified by large language models (LLMs), face challenges in user-facing applications because of their generic nature and lack of personalization. Platforms often serve a diverse user base, where people exhibit a wide range of preferences shaped by their characteristics and interactions with content. This heterogeneity cannot be captured by a generic LLM that imposes a single distribution of generated text on all users. For example, in answering "What are the advantages of the new iPhone," the technical depth and application breadth of the generated answer should be tailored to the user's expertise in technology, and a generated summary of product reviews should emphasize different product aspects based on individual needs. Personalizing LLM responses has therefore become essential to enhancing user satisfaction and content relevance (Woźniak et al., 2024; Zhang et al., 2024; Salemi et al., 2024).

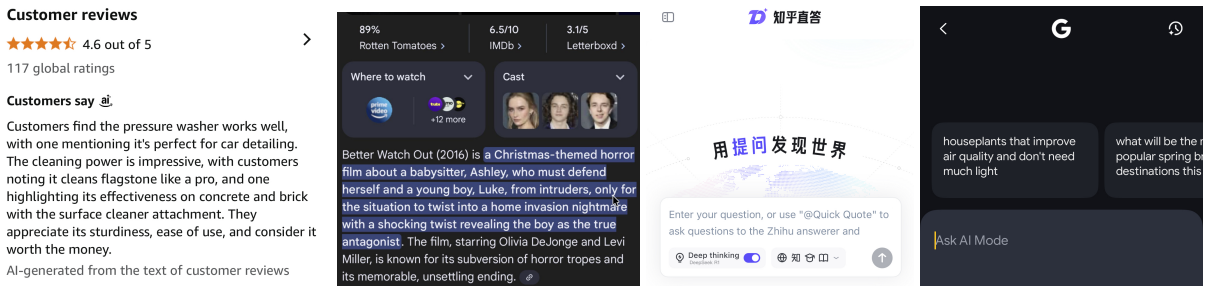


Figure 1: Platforms increasingly generate content for consumers, such as summaries (e.g., for product reviews and movie synopses) and answers (e.g., for questions and searches).

However, personalizing LLMs is often costly under existing approaches. To reveal user preferences, a popular way is to leverage human annotation data, which presents users with multiple candidate responses to elicit pairwise choices or full rankings (Ziegler et al., 2020; Rafailov et al., 2023). Such annotations are

costly to collect, resulting in limited scale, especially at the individual level. In contrast, online content platforms typically estimate preferences from users' historical interactions (e.g., ratings and clicks) using conventional personalization methods such as recommendation systems. The interaction data are large scale, easy to collect, and readily available to many platforms. Yet how to enable LLMs to leverage such interaction data effectively remains underexplored. Alternatively, applications like chatbots often personalize LLMs through in-context learning (ICL), which includes the chat history of a session directly in the system prompt. However, due to high token cost, it often becomes economically infeasible for content platforms to incorporate the extensive content users have interacted with in the prompt. For example, an approximate estimate indicates that the prompt-based approach will lead to a cost of over $200 million per month for the Q&A platform in our application to offer personalized content.[1]

We present Generative Personalized Targeting with Generative Pre-trained Transformers (GPT^2), a framework that generates personalized responses catering to individual preferences. It infers user preferences from both textual content and non-textual interaction data and leverages the inferred preferences to guide text generation. GPT^2 reflects the premise that user preferences drive interactions with specific documents, which in turn exhibit shared linguistic characteristics. The framework couples two components under a unified probabilistic structure. The first component is a user model based on the idea that users with similar interaction patterns with content would share similar linguistic preferences. Rather than assigning each user an independent embedding (Li et al., 2024b), the proposed model captures user dependencies and uses an amortized encoder to generalize to new users. We further design an attention mechanism that allows the user representation to encode different linguistic features like emotion and brevity. The second component is a language model that transforms the user representation into soft prompts residing in the same space as word token embeddings, and prepends them to the prompt text. The generation is then personalized by conditioning on both the prompt text and the soft prompt that contains the user's inferred preferences. The two components are estimated jointly, so the preferences expressed through interactions and revealed in the favored texts together inform a shared user representation without requiring external annotations or self-explication.

We apply GPT^2 to two applications, summarization and question answering, that together represent a broad range of marketing applications of generative models (e.g., those in Figure 1). The first application is generating personalized headlines that summarize news articles, using large-scale news consumption data from Microsoft News. Headlines are short in length, and this compact response space allows model valida-

[1] The estimation is based on statistics in Table 6 and the $2.50/1M input-token price of GPT-4o.

tion based on exact word usage. Our data include a lab-based evaluation set in which held-out participants wrote their own preferred headlines, providing self-elicited benchmarks reflecting genuine individual preferences. Across lexical and semantic similarity metrics, we find that GPT^2 generates headlines closer to users' self-written references than the generic fine-tuned LLM, personalized LLM baseline methods (Li et al., 2024b; Sayana et al., 2024), and non-LLM personalization methods (Ao et al., 2021). We further show that the comparison metrics computed by an LLM-based judge closely track those based on the human-written references, validating LLM-based evaluation in our context.

The second application is question answering on Zhihu, one of the largest knowledge-sharing platforms in the world, where short questions elicit long, open-ended answers that can reflect preferences over a wide range of textual features. We first validate the model in a semi-synthetic study that uses the real question and answer texts but simulates interactions based on given text preferences. In this controlled setting, GPT^2 accurately recovers the underlying preferences and adapts its generation accordingly. Using the observed data and a personalized LLM-based evaluator, we find that GPT^2 consistently achieves higher win rates over the benchmark models, across groups clustered either by the activity level or by the learned preference embeddings. Notably, the gains are most pronounced for low- and medium-frequency users, consistent with the model's ability to share preference information across similar users. We then zero in on the linguistic features that are identified by the model and reflected in its generated texts. Using the Linguistic Inquiry and Word Count (LIWC) dictionaries (Boyd et al., 2022), we identify linguistic features, such as emotional tone, analytical style, and achievement orientation, that differentiate the answers upvoted by each user group. Applying the same dictionaries to the generated answers, we find that over 60% of the discriminating features for random users, and over 80% for active users, exhibit the same directional pattern as in the texts those users actually preferred. This high alignment and wide range of linguistic features provide interpretable evidence of how GPT^2 tailors generation to heterogeneous user needs.

Substantively, we compare *generative targeting*, which creates personalized content for each user from an open-ended choice set, with traditional *discriminative targeting*, which selects from a finite set of existing items. As platforms increasingly use hybrid interfaces to offer personalized content that combines existing content and newly generated material, we leverage GPT^2 and our large-scale, fine-grained individual-level data from Zhihu to uncover the complementarity of the two targeting paradigms. Comparing answers generated by GPT^2 with answers recommended by a discriminative algorithm, we identify the conditions under which generative targeting offers the greatest complementary value for similar platforms, such as when ex-

isting responses are short, when questions require broad knowledge, and when users have limited interaction histories. Moreover, we demonstrate that generative targeting provides a practical solution to new content's cold-start problems (Padilla and Ascarza, 2021), a challenging setting for predictive methods. The probabilistic structure of GPT^2 enables personalized ranking of new responses with few user interactions using the text likelihood. As generated content becomes a routine part of platform assortments, these findings offer guidance on when and how firms could allocate generated and existing content across heterogeneous users.

This article makes several contributions. First, we develop a probabilistic framework that transforms a generic LLM to generate targeted content at the individual level. The framework connects the marketing tradition of consumer behavior modeling to modern generative AI. Second, we improve existing LLM personalization by better generalizing to new users and enabling data sharing among users with similar behaviors, which particularly benefits users with sparse individual history. Third, our method enables platforms to leverage large-scale interaction data to personalize generative AI. Such data remain underexplored in this context, but reflect complex content preferences and impose minimal economic and computational cost under our approach. Fourth, we conduct two empirical studies using large-scale, individual-level data, yielding evidence on much more diverse and realistic user preferences than existing work that relies largely on synthetic personas. Finally, we provide timely insights into achieving complementarity between new generative targeting and conventional discriminative targeting approaches.

The remainder of the article is organized as follows. Following a review of the literature in Section 2, we introduce the modeling framework in Section 3, detailing its probabilistic structure and inference procedure. Section 4 presents the application on personalized headline generation, where we validate the model against human-written references and benchmark methods. Section 5 presents the application on question answering, including a semi-synthetic validation study, benchmark comparisons, interpretable analyses of the learned preferences, and the substantive implications of generative versus discriminative targeting. We conclude in Section 6 with a discussion of contributions, limitations, and directions for future research.

2 Literature

This article relates to and contributes to several streams of literature. We first review the connections to methodological work at the intersection of language modeling and preference measurement. We then position our work within the marketing applications of personalization and targeting.

First, our framework builds on methodological research that jointly models unstructured text and user behaviors. The classic supervised topic models (STMs) jointly fit a generative topic model over documents and a regression model that predicts user responses from the estimated topic assignments (McAuliffe and Blei, 2007; Ansari et al., 2018). Although both approaches leverage text and interaction data, our goal differs from that of STMs. STMs use textual representations to assist in predicting user interactions; we instead use preference representations learned from user interactions to personalize text generation.

Second, our framework extends recent work on personalizing LLMs (Zhang et al., 2024). A common approach is to incorporate user histories into prompts through in-context learning (ICL). Though this approach may work well for settings like chat sessions, directly including a user's full interaction history in the prompt is often infeasible, because rich histories produce long prompts, resulting in high token cost. Some studies therefore personalize at the level of shared group personas (Chen et al., 2024), fine-tune a separate model for each persona (Tan et al., 2024b), or retrieve a small subset of relevant historical records (Li et al., 2024a; Salemi et al., 2024; Shi et al., 2025). While conceptually appealing, retrieval-based methods discard much of the available history, and the resulting personalization is opaque because it depends on the pretrained model's ability to exploit the retrieved context.

To personalize at lower cost, an alternative is to encode user information into compact embeddings that an LLM can leverage as additional inputs. Prior work uses such embeddings for both prediction and generation. For prediction, Ning et al. (2024) and Doddapaneni et al. (2024) encode a user's sequential interactions and integrate the embedding with the LLM through cross-attention or as an additional prompt input, targeting tasks such as predicting document attributes or the next item.

This article instead focuses on generation. Embeddings have been used as soft prompts or prefixes for controllable generation (Li and Liang, 2021; Lester et al., 2021), where they hold the language model fixed and learn a task-level prompt. Li et al. (2024b) apply soft prompts to personalization by assigning each user an independent prompt indexed by user ID. Sayana et al. (2024) adopt a two-stage process that first learns user embeddings from user-product interactions, then fits an adaptor mapping them to soft prompts for a frozen LLM. Because the interactions with products need not reflect preferences for the content and the LLM is not updated, this design yields limited personalization in our applications. Additionally, due to the challenges in evaluating generative models without ground truth labels, much of this literature evaluates generation by simulated data such as synthetic personas and LLM-generated references (Chen et al., 2024; Sayana et al., 2024). To address these gaps in the literature, our proposed method jointly models user

Table 1: Comparison of recent LLM personalization methods.

Study	Goal G/P	Real data	Preference evidence	Method	Train LLM
Liao et al. (2024)	P	Yes	User-product interactions	Embedding	Yes
Zollo et al. (2024)	G	No	Synthetic likes/dislikes	Retrieval, ICL	No
Li et al. (2024b)	G	Yes	Pairwise choice	Embedding	Yes
Choi et al. (2025)	P	No	Pairwise choice	Embedding	Yes
Tan et al. (2024a)	Mixed	Yes[a]	Authored text, tags, reviews, product ratings	LoRA-piece selection, Assemble	Yes
Shi et al. (2025)	Mixed	Yes[a]	Authored text, tags, reviews, product ratings	Retrieval, ICL	No
Richardson et al. (2023)	Mixed	Yes[a]	Authored text, tags, reviews, product ratings	Retrieval, ICL	Yes
Park et al. (2024)	P	Yes	Pairwise choice	Embedding	Yes
Salemi et al. (2024)	Mixed	Yes[a]	Authored text, tags, reviews, product ratings	Retrieval, ICL	Both
Sayana et al. (2024)	G	No	User-product interactions, content text	Embedding	No
Ours	G	Yes	User-content interactions, content text	Embedding	Yes

Notes. G: generation; P: prediction. [a] Uses or introduces the benchmark tasks in Salemi et al. (2024), which are derived from observed historical records instead of elicited preferences.

behavior and language, learns compact embeddings from user-content interactions, and conducts comprehensive evaluations through semi-synthetic data, validated LLM evaluators, linguistic feature analysis, and lab-based human-written references. Table 1 provides a qualitative comparison with related work on LLM personalization across dimensions such as goals, data use, and methods.

Third, our work relates to LLM post-training methods designed to align model outputs with human preferences. Popular methods such as Reinforcement Learning from Human Feedback (RLHF, Ziegler et al., 2020) and Direct Preference Optimization (DPO, Rafailov et al., 2023) generally capture population-level rather than individual preferences, and require costly annotation data (Zhao et al., 2025). Our framework is able to adapt to these post-training approaches and turn them into personalized generation when proper annotation data are available (e.g., an example shown in Web Appendix VIII).

Finally, an extensive marketing literature studies targeting and personalization, seeking efficient strategies to tailor marketing actions to individual consumers (Rafieian and Yoganarasimhan, 2023; Lemmens et al., 2026). For instance, recommender systems suggest targeted products that maximize predicted outcomes like click probabilities, and dynamic models adapt actions to a user's evolving behaviors (Misra et al., 2019; Liu, 2023). These policies are discriminative in nature, using prediction to select from a finite set of existing alternatives. Recent work leverages LLM outputs as prior estimates, explanations, and reasoning to improve the predictive accuracy (Wang et al., 2025a; Ye et al., 2025; Wang et al., 2024). In this article, we

contribute to this stream by analyzing the proposed generative targeting framework that creates personalized content beyond the predefined choice set, and we study how it complements conventional discriminative targeting systems.

3 Modeling Framework

Consider the setting where we observe N users indexed by $i \in \{1,\ldots,N\}$, D text prompts x_d for $d \in \{1,2,\ldots,D\}$, and J response texts y_j for $j \in \{1,\ldots,J\}$. Each prompt can have either multiple observed responses or only one response. The texts x_d and y_j are represented as tokenized sequences, where each token at position t is encoded from a word or phrase. Each token is an integer $x_d^t, y_j^t \in \{0,1,\ldots,V\}$, where V is the vocabulary size. We denote the lengths of the tokenized sequences as $|x_d|$ and $|y_j|$.

In addition to the textual data, each user i has an interaction history set $\mathcal{H}_i \subset \{1,\ldots,J\}$ over the textual responses. For each text $j \in \mathcal{H}_i$, we observe the interaction r_{ij} of user i with this text. For example, the prompt and response could be news body and headline for creative content summarization or question and answer texts on a Q&A platform. Correspondingly, the interactions could be clicks, upvotes, or downvotes.

3.1 Background

We first review the standard framework for generic LLMs and formalize the consequences of this generic generation for a heterogeneous consumer base. An LLM is a probabilistic model $\texttt{LM}_\theta$ with a large set of parameters θ, primarily based on multi-layer Transformer architectures (Vaswani et al., 2017; Brown et al., 2020). It autoregressively predicts the probability of the next token in a document, conditioning on both the prompt tokens and the previously generated response tokens. Specifically, an LLM models the probability that the t-th token of response j takes the value y_j^t, denoted as $\texttt{LM}_\theta(y_j^t \mid y_j^{<t}, x_d)$, given the preceding sequence $y_j^{<t}$ and the prompt sequence x_d. Each token is represented as an L-dimensional word token embedding vector. An LLM is pre-trained on a diverse, large-scale corpus and then adapted with a relatively small amount of domain-specific data during post-training. This adaptation is commonly performed through supervised fine-tuning (SFT), which maximizes the likelihood of the domain-specific data by $\max_\theta \sum_{(x_i,y_i)} \log \texttt{LM}_\theta(y_i \mid x_i)$.

However, this generation process is user-agnostic, imposing the same response distribution $\texttt{LM}_\theta(y_j \mid x_d)$ on all users and neglecting individual heterogeneity. As a consequence, the generated responses are more likely to resemble the responses that appear most frequently in the fine-tuning data, a statement we formalize

as Proposition 1 in Web Appendix I. For example, consider the user question "What are the advantages of the new iPhone." If technical answers appear more often than non-technical ones in the post-training data, a generic LLM will tend to generate technical answers even for users who prefer non-technical answers. This majority-driven behavior underscores the need to personalize LLMs to capture users' unique preferences and generate targeted responses.

3.2 Probabilistic Model of GPT^2

We propose GPT^2 as a probabilistic model that integrates language modeling and user modeling to enable personalized generation. Its user model component constructs user embeddings from observed interactions with text documents, capturing the preferences reflected in user behaviors. Its language model component then links the user embeddings to the preferred textual content and captures multifaceted linguistic preferences. Figure 2 illustrates the overall model structure.

3.2.1 User Model

The user component models the user reactions to texts by encoding the user history into user-specific parameters. A traditional mean-field approach uses one individual parameter for each user (Li et al., 2024b), which only learns embeddings for the users in training set. To generalize to new users, our user model adopts an amortized approach that learns a user embedding from the history set with a shared encoder.

Next, instead of summarizing the document embeddings in the history $\mathcal{H}_i$ by simple mean pooling, as commonly used in standard deep set methods (Zaheer et al., 2017), we further design an attention mechanism that allows each latent factor to attend to a potentially different slice of the user's history. This approach enables different latent factors to capture user patterns related to specific linguistic features.

Formally, we model each document j with a latent embedding vector $\beta_j \in \mathbb{R}^K$, where K is the embedding dimension. For each user i and their interaction history $\mathcal{H}_i$, we compute

$$\eta_{ijt} = w_t^\top \phi(\beta_j) \quad \text{for } j \in \mathcal{H}_i \tag{1}$$

$$a_{ijt} = \frac{\exp(\eta_{ijt})}{\sum_{j' \in \mathcal{H}_i} \exp(\eta_{ij't})}, \tag{2}$$

$$h_{it} = \sum_{j \in \mathcal{H}_i} a_{ijt} \phi(\beta_j), \tag{3}$$

$$z_{it} = \rho(h_{it}). \tag{4}$$

Here, $\phi(\cdot), \rho(\cdot) : \mathbb{R}^K \to \mathbb{R}^K$ are two fully-connected neural networks for flexible transformation, and $w_t \in \mathbb{R}^K$ denotes the t-th column of a learnable attention matrix. η_{ijt} and a_{ijt} are the attention logits and weights that link the j-th interacted document to the t-th soft prompt token of user i. Finally, z_{it} is the latent factor that is jointly associated with user i's behavior pattern and language preferences.

We compute the user embedding $\alpha_i \in \mathbb{R}^K$ by aggregating the latent factors as

$$\alpha_i = \frac{1}{T} \sum_{t=1}^{T} z_{it}. \tag{5}$$

We then link each user embedding α_i to the behavior pattern of this user. For a user-content pair (i, j), the observed behavior r_{ij} is then modeled as

$$r_{ij} \sim p(r_{ij} \mid i, j) = \text{Categorical}\big(\sigma\big(f(\alpha_i \oplus \beta_j)\big)\big), \tag{6}$$

where $\oplus$ represents vector concatenation, $f(\cdot)$ is a feed-forward classifier and $\sigma(\cdot)$ is the softmax function. For example, $f(\cdot)$ can output the logits for clicks and non-clicks or for upvotes and downvotes on the response documents, where r_{ij} can be dummy-coded as the type of reactions. Equation (6) is based on neural collaborative filtering (NCF; He et al., 2017), which captures nonlinear patterns. Our user model extends NCF by amortizing embeddings over interaction histories and by the attention mechanism that lets latent factors z_{it} encode multifaceted linguistic preference to further guide language generation. For computational efficiency, we model user interactions based on the non-textual data. We find it achieves a high predictive accuracy of over 90% in our applications, and further incorporating textual data and metadata in the user component yields only marginal improvements, though the proposed framework supports flexible alternative modeling for user interactions.

3.2.2 Language Model

The language model component uses the learned latent factors from the user model to construct the soft prompt for guided response generation. Soft prompts are learnable vectors that an LLM can take as input. For LLMs to understand a soft prompt, its tokens need to reside in the same embedding space as the embeddings of word tokens. We create the soft prompt from the latent factors z_{it} rather than from the aggregated embedding α_i that would lose information from aggregation. The soft prompt is constructed to include multiple tokens that correspond to distinct aspects of linguistic preference.

Specifically, let $h_\Pi(z) : \mathbb{R}^K \to \mathbb{R}^L$ be a function that maps latent factors $\{z_{it}\}$ to the LLM embedding

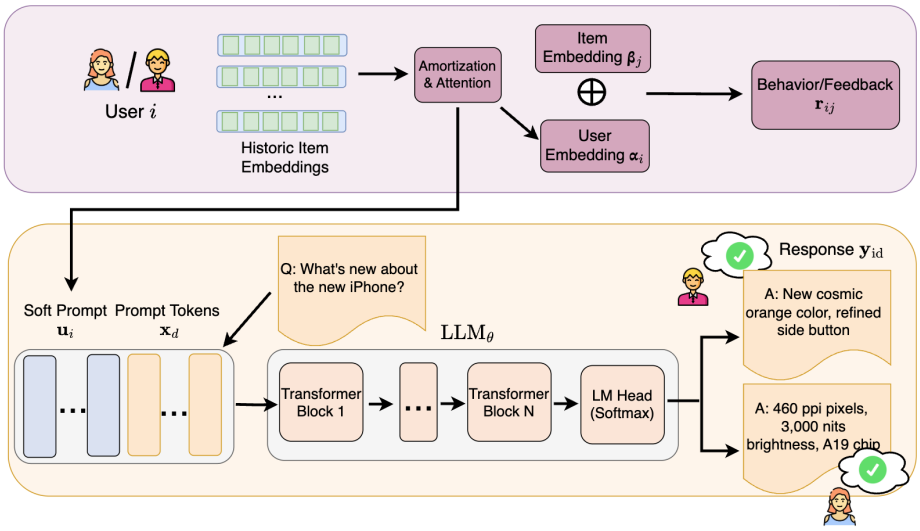


Figure 2: Method overview of GPT^2 in generating personalized responses.

space, where Π are the coefficients, and L is the token-embedding dimension of the base language model (e.g., L is 4,096 in our application). For each user i, we define

$$u_{it} = h_\Pi(z_{it}), \qquad t = 1, \ldots, T, \tag{7}$$

for each token t, and the soft prompts for user i are collected in $U_i = [u_{i1}, \ldots, u_{iT}]$. For instance, $h_\Pi(z)$ can be a fully connected neural network or a linear mapping $h_\Pi(z) = \Pi z$ with projection matrix $\Pi \in \mathbb{R}^{L \times K}$. The mapping leverages an essential technique in learning soft tokens by starting in a low-dimensional space, then transforming to the high-dimensional word token embedding space (Li and Liang, 2021).

We next prepend the soft tokens U_i to the embedding sequence of the prompt tokens and treat the extended sequence as the input to the language model. The personalized response $y_{id} = (y_{id}^1, \ldots, y_{id}^{M_{id}})$ for a prompt x_d is generated as

$$y_{id} \sim p_\theta(y_{id} \mid x_d, i) = \prod_{m=1}^{M_{id}} \text{LM}_\theta\left(y_{id}^m \mid y_{id}^{<m}, x_d, U_i\right). \tag{8}$$

The response y_{id} depends on the soft prompt tokens U_i, which makes it conditioned on a specific user i and thus achieves personalization. The user preferences, expressed through their interactions with content, are captured by the shared latent variable $\{z_{it}\}_{t=1}^T$ through the user model, which simultaneously guides the LLM to generate texts that align with the preferred semantic content and style. The synergy between the user and language components arises when users' interactions are driven by linguistic preferences. This

implies that interactions should be with textual content, as interactions with products alone may not provide information useful for guiding the generation of texts.

In addition, since the pre-training of an LLM uses a finite vocabulary list (e.g., 50,400 in our applications), the vocabulary occupies only a subset of the full space $\mathbb{R}^L$. To prevent the soft prompt tokens from extrapolating beyond the support of the LLM, we propose a language-encouraging prior. Specifically, instead of random initialization, we initialize the columns of the projection matrix Π with the embeddings of randomly selected words from the vocabulary. This prior ensures that the initial u_{it}, which is a linear combination of the Π columns, lies within the word space spanned by the word token embeddings. Finally, since the item variables are not used as inputs to the language model, we place a noninformative prior on them: $p(\beta_j) \propto 1$.

3.2.3 Joint Estimation

Taken together, the user and language components couple the user preferences expressed through explicit or implicit interactions with the preferences revealed in the favored textual content. The user model encourages users with similar feedback patterns to share similar representations z_{it}, which further produces similar soft prompts.[2] Thus, it enables the language model to learn targeted generation for user i by fine-tuning on the responses preferred by this user and by users with similar preferences.

We denote the attention parameters $\{w_t\}_{t=1}^T$ and neural network parameters of (ϕ, ρ, f) for the user model collectively as θ_u, and the language model parameters as θ_L. We infer the latent variables $\{\beta_j\}$ and model parameters (θ_u, θ_L) by maximum a posteriori (MAP) estimation. Let $\mathcal{H}_{id} \subset \mathcal{H}_i$ be the response texts for prompt x_d that were interacted with by user i. The joint distribution of the data and variables is

$$p_{\theta_u,\theta_L}(x,y,r,\beta) = \left[\prod_{d=1}^{D} p(x_d) \prod_{i=1}^{N} \prod_{j \in \mathcal{H}_{id}, r_{ij}=1} \prod_{t=1}^{|y_j|} \mathtt{LM}_{\theta_L}(y_j^t \mid y_j^{<t}, x_d, U_i)\right] \times \left[\prod_{i=1}^{N} \prod_{j \in \mathcal{H}_i} p_{\theta_u}(r_{ij} \mid \mathcal{H}_i, \beta_j)\right] \prod_{j=1}^{J} p(\beta_j). \tag{9}$$

The language model is fine-tuned on the texts that are positively interacted with by the users such as being clicked or upvoted and labeled as $r_{ij} = 1$. The MAP estimation objective is $\max_{\{\beta_j\},\theta_u,\theta_L} \log p_{\theta_u,\theta_L}(\beta \mid x, y, r)$, which equals the logarithm of the joint distribution in Equation (9) up to an additive constant. Following the language model training convention, we normalize the log-likelihood of the response tokens by the

[2] This is because $\|u_{it} - u_{i't}\|_2 \leq \|\Pi\|_{\text{op}} \|z_{it} - z_{i't}\|_2$ with operator norm $\|\cdot\|_{\text{op}}$.

response length as $-(1/|y_j|)\log\prod_{t=1}^{|y_j|}\texttt{LM}_\theta(y_j^t \mid y_j^{<t}, x_d, U_i)$ (i.e., the log-perplexity of the response; Dhillon et al., 2025). A small amount of stochastic dropout is applied to the items in the history during training to prevent the attention encoder from overfitting.

We adopt the filtered supervised fine-tuning (f-SFT) as the post-training paradigm, following Li and Liang (2021); Dong et al. (2023). Instead of fine-tuning on all responses to encode domain knowledge, we select high-quality responses receiving positive feedback and fine-tune the model on these filtered samples. Unlike traditional f-SFT that filters samples by an automatic, generic reward model (Dong et al., 2023), our sample selection is based on real human feedback and reflects diverse individual preferences. This approach has two additional major benefits for personalization. First, it is more flexible than alternative off-policy preference learning approaches that require multiple responses to a single prompt or additional preference labels. Second, the token-level likelihood provides a dense signal for capturing subtle language patterns, compared to a sparse reward (e.g., pairwise choice) associated only with the overall response. Nevertheless, our framework adapts to various post-training techniques; for example, when a prompt has multiple responses labeled as preferred or not, we can fine-tune the LLM using Direct Preference Optimization (Rafailov et al., 2023) to generate personalized content. Web Appendix VIII discusses this extension in detail and provides empirical validations.

In the following sections, we present two complementary empirical studies that provide individual-level textual responses and the observed non-textual interactions. These fine-grained observations enable detailed analysis of the benefits of personalization with our proposed generative model. The first application is on news content summarization with long prompts and short responses. The compact response space allows for word-level comparisons. The second application is on question answering on a digital platform with short prompts and long responses, where the large response space allows the generated text to reflect individual preferences across a wide range of linguistic features.

4 Application 1: Summarization of News Content

Summaries of online content have drawn growing attention in marketing as they help consumers decide which products to consume (Toubia, 2021). Major platforms have begun deploying LLMs for summarization, including Amazon for product review summaries and Netflix for content synopses. In this application, we study the summarization of online news articles into headlines, which condense article content into con-

Table 2: Descriptive statistics of the Microsoft News data set.

		Min	Median	Mean	Max
Observed Data for Training	Headline length (words)	2	11	11.33	40
	Body length (words)	11	546	838.65	3,000
	Clicks per user	36	43	43.59	60
Human Evaluation Set for Testing	Clicks per user	3	14	17.05	64
	Written headlines per user	23	32	32.77	46

cise and informative wording. Headlines that are tailored to individual preferences can help readers identify relevant information more efficiently and increase user engagement (Banerjee and Urminsky, 2025). The application leverages a large-scale corpus of news articles from Microsoft News. Our focus in this application is on validating the framework against human evaluation data and providing interpretable evidence on the personalization effect.

4.1 Data Set Description

The data set contains user click behavior on news headlines, together with the textual content of the news articles, from Microsoft News between June 14 and July 12, 2019 (Ao et al., 2021). Each user is exposed to a set of news headlines in an impression, and the impression logs record the IDs of the clicked and unclicked news items from this set. This data set captures rich, real user behavior patterns and heterogeneous preferences that simulations, such as those based on synthetic personas, typically cannot provide.

For computational feasibility, we select a subsample of users and news articles, focusing on relatively active users. Specifically, we select users whose number of news exposures is between the 60th and 80th percentiles in the data and who have balanced distributions of clicks and non-clicks. The procedure prevents outlier users from strongly influencing the data, as the number of news items exposed to each user follows a long-tailed distribution, with a maximum of 1,890[3]. This results in a large set of 2,743 users, with 90% randomly selected for model training, 16,094 distinct news articles, 212 thousand interactions, and around 100 million tokens from the text content. For the textual data, we use the text of all the clicked headlines and their news bodies, which include 106,498 records of user-news pairs. For the non-textual interactions, each user clicked 44 articles on average. The descriptive statistics are summarized in Table 2.

More importantly, the data contain a separate human evaluation set that complements the observed be-

[3] With sufficient computational resources, our model can incorporate more data, and users can be reweighted by click frequency.

havioral data described above. This evaluation set was collected in two stages from 103 English-speaking participants, recruited independently of the Microsoft News users in model training data. In the first stage, each participant browsed hundreds of news headlines and clicked on headlines of interest, yielding 17 clicked headlines per participant on average. In the second stage, each participant was asked to write a preferred headline for an average of 33 articles without being shown the articles' original headlines. The articles used in this stage were excluded from the first stage. The quality of these manually-written headlines was checked by professional editors. Thus, these headlines faithfully reflect self-elicited preferences regarding the qualities of a good headline for each participant (Ao et al., 2021). We use this set of 3,375 articles and their self-written headlines for evaluating our methods.

Several characteristics make this application well-suited and valuable. First, the data set includes both user interaction data and news text, whereas many existing data sets include only interaction data for training predictive models or only text for training language models. Second, the short headlines limit potential textual variation, enabling comparison and validation based on word usage. Third, each news body is associated with a single headline, a setting that highlights the applicability of our method when approaches that require contrasting multiple responses (Rafailov et al., 2023; Zhao et al., 2025) are not feasible. Finally, the meticulously collected human evaluation data provide personalized reading interests and preferences, serving as a benchmark for comparing generations from different methods.

4.2 Model Validation

We validate whether $\texttt{GPT}^2$ can generate headlines better reflecting individual preferences than existing approaches. We describe the benchmark methods, and then evaluate generation quality using users' self-written headlines, as well as an LLM-based evaluator.

4.2.1 Benchmark Methods

We validate the model performance by comparing it with three related approaches. The first benchmark model, denoted as *Generic SFT*, collects all the clicked articles and headlines and fine-tunes the pre-trained LLM on these documents using supervised learning. Through this refinement, Generic SFT learns the context of the task as summarization and captures user preferences from the positively interacted headlines. However, Generic SFT does not vary the distribution of generated headlines across users, thus it serves as a benchmark for examining the effect of personalization.

The second benchmark method, denoted as *PSFT*, is adapted from Li et al. (2024b) as an embedding-based personalization method for LLMs. PSFT assigns each user an embedding variable, which is projected into soft prompts, as in GPT-2. However, PSFT leverages only the text data and does not personalize based on users' observed interaction behaviors. As a result, its user embeddings are independent, so one user's preferred response texts do not influence another user's embedding. Comparison with PSFT reveals the benefits of learning from non-textual interaction data.

In addition, we compare against the fusion architecture of Sayana et al. (2024) and denote it as *REGEN*. It encodes each user-item interaction in two decoupled stages. The first stage learns embedding vectors for each user by collaborative filtering based on users' clicks or ratings. The second stage learns an adaptor that transforms the user embeddings to the LLM input space. REGEN does not update LLM parameters, and its user embeddings are not adapted for text generation, as they are held fixed in the second stage. We use this benchmark to show the essential role of joint learning and the attention-based user modeling in capturing individual preferences.

For all methods, we use a pre-trained LLM with 6 billion parameters (GPT-J; Wang and Komatsuzaki, 2021).[4] We report the computation details of the benchmark methods in Web Appendix II.

4.2.2 Main Evaluation Results with Reference Headlines

We first evaluate how well the generated headlines align with individual preferences by comparing them to the users' self-written headlines. Because each user is instructed to write headlines reflecting their own preferences, the similarity between the generated and self-written headlines provides a direct measure of this alignment.

We use two sets of similarity metrics. The first measures word overlap between generated headlines and users' preferred headlines, which serve as references. We report the F1 scores of ROUGE-1 and ROUGE-L (Lin, 2004), which measure unigram overlap and the longest common subsequence, respectively, between generated and reference headlines. We further report BLEU scores that measure n-gram precision with a brevity penalty (Papineni et al., 2002). These scores reward generations that preserve local lexical patterns from the reference. Since ROUGE and BLEU capture only surface-level lexical overlap, we also evaluate the BERTScore, which computes similarity between contextual embeddings of the generated and reference

[4] We use GPT-J 6B as it shares the core architecture of modern LLMs with 28 layers of Transformer decoder blocks. Because our approach operates on the input representations rather than the model weights, it transfers directly to other decoder-only architectures such as Llama and Gemma.

Table 3: Evaluations of targeted headline generation quality.

Method	ROUGE-1	ROUGE-2	ROUGE-L	BLEU-1	BLEU-2	BLEU-4	BERTScore
Pointer-Gen	19.86	7.76	18.83	–	–	–	–
LSTUR	23.71	8.73	21.13	–	–	–	–
NAML	28.01	10.72	22.24	–	–	–	–
REGEN	18.40	5.09	15.58	14.05	5.41	0.73	3.71
Generic SFT	28.99	10.35	23.86	22.38	11.14	2.21	20.31
PSFT	29.44	10.67	24.40	22.60	11.37	2.39	20.70
GPT^2	**30.51**	**11.77**	**25.20**	**23.52**	**12.39**	**2.95**	**21.31**

Notes. The scores are percentages. The BERTScore rescales with default baseline in the Python package bert-score.

headlines and captures semantic agreement even when the wording differs.

Table 3 shows the evaluation scores. We additionally report the results of Pointer-Gen, LSTUR, and NAML from Ao et al. (2021), which represent the state of the art in non-LLM-based personalized headline generation.[5] We find that GPT^2 consistently achieves the best score across all metrics. REGEN has the lowest performance, suggesting the necessity of jointly modeling user behaviors and text content, and post-training LLMs. Relative to PSFT, GPT^2 improves ROUGE and BLEU scores, showing better preservation of local word and phrase patterns from the user-written reference. The higher BERTScore suggests stronger semantic agreement with the preferred headline. The improvement reveals the benefits of our user modeling, which captures the dependence among the users. This dependency structure enables users with similar interaction patterns to share the textual data of the interacted headlines through similar soft prompts.

The gap between GPT^2 and Generic SFT is large for both lexical overlap and semantic similarity. This suggests that our generation achieves personalization to diverse user tastes. Though we use a relatively small LLM with 6 billion parameters, we note that the post-trained LLMs generate headlines closer to user needs than the best non-LLM approaches. Given scaling laws, this advantage might be more salient as LLM size increases. Therefore, personalization with the generative model has a strong potential to target users with highly personalized content from an unbounded assortment.

4.2.3 Evaluation by Judge Models

Our data include a valuable component for human evaluation, however, collecting individual preference data is challenging in general, as it requires collecting both historical actions and preferred content for

[5]These methods were calibrated on a larger data set than our sample and evaluated on the same test users.

each user. With advances in LLMs, researchers have been using frontier LLMs as evaluators for assessing generated text quality (i.e., "LLM-as-judge") (Gu et al., 2024). This approach is flexible across application contexts and has demonstrated high agreement with controlled and crowdsourced human preferences (Zheng et al., 2023). We first adapt an LLM-based evaluator to our headline generation context and then compare it with the human evaluation.

For each news body in the test set, we compare the headlines generated by two methods through a pairwise choice made by OpenAI's GPT-4o model (OpenAI, 2024), a widely used and validated model for LLM judging (Rafailov et al., 2023; Gu et al., 2024). The LLM evaluator determines whether one model's generation is a win, loss, or tie compared with that of the other model. We then compute the fraction of wins in the test set as the instance-level *win rate*. For each user, we also compute a user-level win rate, where a model is counted as winning if the judge prefers its headlines on a majority of that user's test news instances. To reflect personal preference in evaluation, we follow Li et al. (2024b) and Singh et al. (2025) to include all the historically clicked headlines of a user in the prompt and ask the LLM to reason about the preference patterns of this user. We adopt the AlpacaFarm protocol (Dubois et al., 2023), a set of implementations that is designed to improve robustness to the annotator configurations and mitigate known biases in LLM-based evaluation such as position and length.

Table 4: Pairwise preference results for $\texttt{GPT}^2$ versus PSFT.

Source	Instance-level win rate	User-level win rate
LLM Judge	0.5159	0.5667
BERTScore	0.5151	0.5806
Score-based Majority Vote	0.5180	0.6092

The results are shown in Table 4. We further compute a reference-based win rate, in which a model wins on an instance if it attains a higher similarity score to the human-written reference headline than the comparison model. We report this win rate under the semantic similarity metric BERTScore, and under a majority vote across all metrics reported in Table 3, where a model wins if it scores higher on a majority of them.

From these results, $\texttt{GPT}^2$ has a higher win rate over PSFT according to the LLM evaluator. We note that the user-level win rate is generally higher than the instance-level win rate. This is likely because the user-level evaluation has fewer ties and hence reduces instance-level noise. The higher user-level win rate suggests that the user model of $\texttt{GPT}^2$ accurately captures the preferences of each user. We further

find a consistent evaluation between the LLM evaluator and the similarity-based metrics based on human preferences. The win rate from the LLM evaluator is close to that from the similarity scores, with a gap of less than 0.3% for the instance-level win rate and less than 4.3% for the user-level win rate.

Table 5: Demonstrative examples of the personalization effect.

Original Headline:	The 14 Easiest Ways to Stick to Intermittent Fasting
GPT^2 for User A:	14 Subtle Ways You're Probably Not Eating Intermittently Fast Enough
User A clicked news:	1. 650 seats, 46 million voters : The UK election in numbers
	2. 15 Things You Should Never Do In Public
	3. Convicted Greenville County Sheriff Will Lewis sentenced to one year in prison
	4. The 12-year-old virtuoso drummer you've probably seen on TV is from N.J.
	5. 8 Things that Can Happen When You Don't Treat ADHD
	6. Big Ten Week 10 power rankings
	7. Granny Pods and 7 More Clever Ways to Earn Income From Your Property
	8. Humane Animal Rescue Waiving Adoption Fees This Month For Pets 7 Years And Older
GPT^2 for User B:	How to Stick to Intermittant Fasting Without Feeling Like a Bear in a Cave
User B clicked news:	1. Car Slams Into Box Truck On Cone Boulevard: Greensboro Police
	2. Japan tax agency finds Ghosn used Nissan money for private use: report
	3. Analysis shows most wrong-way drivers in Arizona are sober
	4. Suspect nicknamed "Woo Woo" charged in three murders, claims to have committed five more: "I want to be a serial killer"
	5. State Court Bars Using North Carolina House Map in 2020 Elections
	6. Sources say first engine broke down en route to Woolsey fire, which grew at a terrifying rate

4.3 Interpretable Results

Next, we illustrate how the proposed generative model captures nuanced linguistic preferences with several demonstrative examples. Table 5 shows the generated headlines for two users about a news article on intermittent fasting. We list the headlines in the clicked history of each user that are relevant to the style of the generated headline; the full list of clicked headlines is provided in Web Appendix III. In this case, user A has a click history with repeated numbers and list-style words such as "15 Things", "7 More Clever Ways" and "Week 10", and the generated headline follows that same taste pattern by turning the article into a numbered listicle: "14 Subtle Ways You're Probably Not Eating Intermittently Fast Enough."

User B, in contrast, has clicked headlines that are much more plainspoken and action-verb-driven, open-

ing in a conventional news style with phrases such as "Car slams into", "Analysis shows", and "Sources say". Interestingly, the generated headline for this user avoids the numbered listicle framing and instead becomes a more direct explanatory sentence, "How to Stick to Intermittant Fasting Without Feeling Like a Bear in a Cave." The two generated headlines therefore preserve the same article topic but shift the presentation in ways that are consistent with each user's own style preferences.

Web Appendix III shows another example where the news content is about local crimes. One user preferred headlines with clear sources such as "AP Interview:" and "Report:", while another user clicked more subject-first headlines with a named person or organization at the start, such as "A.J. Hinch" and "Chinese Tesla rival Xpeng." For the first user, our model generates the headline "Police: Impersonating officer arrested after attempted stop," which uses a similar wire-style framing; for the second user, it generates "Timothy Trivitt, impersonates police officer, charges say," which leads with a specific name. The two generated headlines therefore shift the presentation toward the personalized styles.

The breadth of users' latent linguistic preferences demonstrated by these examples highlights the importance of generative models for personalization. Human-written headlines, though well crafted, cannot simultaneously satisfy readers with such a range of stylistic tastes. Conventional personalization, which often selects from a finite pool of candidate items using recommendation algorithms, is constrained by what has already been written. The proposed generative personalization thereby opens opportunities to relax these constraints, allowing platforms to better tailor offerings to individuals and improve content diversity.

Furthermore, we analyze how the personalization effect by $\texttt{GPT}^2$ converts to economic gains measured by the click-through rate. Specifically, since the user-written headlines reflect their own preferences for text when compared with other headlines, we fit a choice model using sentence embeddings to represent utility and the Bradley–Terry model to represent choices (Bradley and Terry, 1952). We find that the generative personalization potentially improves the click-through rate across all the baseline methods, for example, by 0.46 to 2.12 percentage points compared with Generic SFT; Web Appendix III reports the detailed results.

5 Application 2: Question Answering on Digital Platforms

Although the compact space of summaries enables word-level evaluations, their brevity may be insufficient to reveal rich advantages of generative personalization. The proposed $\texttt{GPT}^2$ is widely applicable to various types of content. In this section, we apply it to question answering, an application that has become

increasingly influential on major platforms, such as Google AI Mode for searching and Zhida on Zhihu for knowledge-sharing.[6] In this application, we address questions including (1) whether the model can generate content aligned with users' known preferences for a specific text feature, (2) how the model achieves personalization internally, (3) what linguistic features it captures, and (4) what advantages generative personalization offers over conventional discriminative personalization.

5.1 Context and Data

Zhihu is the largest knowledge-sharing platform in China, with key features similar to those of Reddit and Quora. Zhihu users can seek and share knowledge, expertise, and experiences by asking questions and contributing answers. A question may receive multiple answers, and users can express a like or dislike for an answer by clicking "upvote" or "downvote"; Figure 3 shows an example.

Table 6: Descriptive statistics of the Zhihu data set.

		Min	Median	Mean	Max
Training	Question length (words)	2	13	14.21	83
	Answer length (words)	1	154	298.09	5,020
	Interactions per user	3	61	139.69	4,195
Testing	Question length (words)	3	13	15.38	81
	Answer length (words)	1	158	303.56	4,980

The data set contains rich information on user activities and textual content from Zhihu's inception in January 2011 to January 2012. For computational feasibility, we consider a random subset of users and exclude questions that receive only a small number of interactions. This results in 2,752 users, 55,562 unique answers across 4,960 questions, and 384,414 user-answer interactions, among which 500 random questions are held out as the test set; see summary statistics in Table 6 and further data details in Web Appendix V.

The user interaction logs contain timestamped voting activities of all registered users. We dummy-coded user-answer interactions as 1 for upvotes, -1 for downvotes, and 0 for exposure without voting, representing positive, negative, and neutral-to-negative behaviors, respectively. For the textual data, we treat the question text as the prompt and the answer text as the response. The upvoted answers contain an average of 298 words, with a median length of 154 words. Figure W2 in Web Appendix V shows the distributions of the number of answers upvoted per user and the word counts of upvoted answers, which indicate substantial

[6] https://google.com/aimode; https://zhida.zhihu.com



Figure 3: Example of a question and its answers.

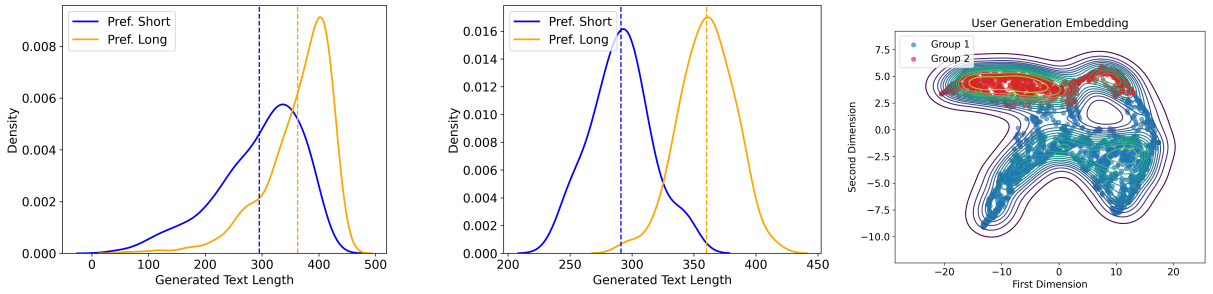
heterogeneity in user activity and answer preferences.

5.2 Semi-synthetic Study

We start by validating the model's ability to capture heterogeneous preferences in open-ended answers through a controlled semi-synthetic study. This addresses the challenge that the true underlying user preferences are not directly observable in the secondary data. The semi-synthetic data use the real questions and answer texts while simulating users' voting behaviors.

Specifically, we simulate two user groups with distinct preferences: among their interacted questions, one group upvotes long answers with high probability (Long-preferring group), and the other group upvotes short ones more frequently (Short-preferring group). Long answers are defined as those with lengths between the 60th and 80th percentiles of all answers, thereby avoiding outliers, and short answers as those below the 20th percentile. Answer length is a key feature that influences user perceptions and can be quantified without ambiguity. We then test whether the model customizes answers to the 500 held-out questions.

Figure 4 compares the generated length distributions for the two simulated groups. In the left panel,



(a) Randomly sampled users. (b) Active users. (c) Distribution of soft prompts.

Figure 4: Panels (a) and (b): comparison of answer length distributions for the users from two distinct groups; panel (c): visualization of the soft-prompt distribution.

users from each preference group are randomly assigned a test question. Answers generated by GPT^2 for the Long-preferring group are substantially longer than those for the Short-preferring group (means of 361 versus 292 words, $p < 0.001$), and answers under 150 words appear only for Short-preferring users. The middle panel restricts attention to the 50 most frequent voters in each group, who provide more informative preference signals. There, the distributional overlap between the two groups is markedly smaller, indicating that GPT^2 personalizes more precisely with more individual interactions. Together, these results show that GPT^2 accurately captures the underlying linguistic preferences and successfully generates responses aligned with them for both randomly sampled and active users.

We further examine the mechanism underlying heterogeneous generation by analyzing the intermediate variables. The right panel of Figure 4 visualizes the distribution of U_i across users, compressed to two dimensions via PaCMAP (Wang et al., 2021). The distribution is clearly bimodal. After clustering, 1,111 users (98.3%) in the first cluster prefer long answers, while 1,345 users (86.5%) in the second prefer short answers. This distinct segmentation suggests that users with similar preferences are grouped closely in the embedding space, and the difference in generated answer length is driven by the preference patterns captured in the soft prompts as intended.

Moving to the individual level, we compare generations for the 50 most active users in each preference group, paired by frequency rank. Each pair is randomly assigned questions from the 500 test questions, so both users in a pair receive the same questions and an average of 10 generated answers each. Figure 5 reports the mean and standard deviation of answer lengths per user. For nearly all pairs, generations for the Long-preferring user are substantially longer than for the Short-preferring counterpart. The results show

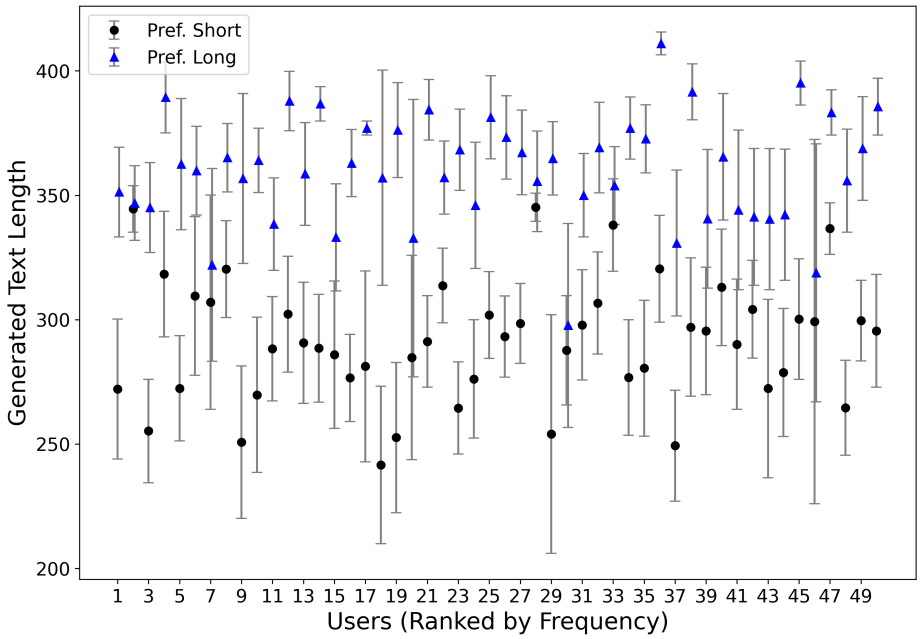


Figure 5: Comparison of answer lengths for individual users in the two groups.

that GPT^2 tailors output length to personal preferences precisely at the individual level. The length intervals also overlap less for pairs with more voting actions, indicating better personalization with richer user data.

The studies with semi-synthetic data provide controlled user preferences and validate how GPT^2 captures users' textual preferences and reflects them in generation. Next, we turn our attention to modeling real interactions and the text corpus on Zhihu.

5.3 Benchmark Comparisons

Generating answers represents an application with short prompts and open-ended responses. Consistent with Section 4, we compare GPT^2 with Generic SFT, PSFT, and REGEN as the benchmark models and use the 6-billion-parameter LLM for post-training. Our focus here is on evaluating the benefits of personalized generation and the joint modeling framework, instead of achieving state-of-the-art generation, which could be largely driven by pre-training data sizes due to the scaling law (Kaplan et al., 2020). In Web Appendix IV, we conduct an ablation study with varying model structures that shows steady performance improvements with increasing model size.

Answer texts are substantially longer than headlines (e.g., 298 words versus 11 words on average), so it is challenging to collect both individual interactions and annotations over many answers in a limited time window as in the news summarization application. We therefore use LLMs as evaluators to provide holistic

assessments as surrogates, an approach we found to align well with human evaluators in Section 4. Such evaluation is suitable for evaluating answer text, as it synthesizes multiple linguistic features like the tone, style, content, and relevance beyond specific word usage. To personalize the LLM evaluator, we find the upvoted questions and answers most similar to the test question and include them in the prompts. See Web Appendix V for details of the evaluator prompts.

We assess how the models perform across heterogeneous user groups. We group the users into high-, medium-, and low-frequency groups based on their number of upvoted answers, or into three preference groups (denoted G1, G2, G3) clustered by their soft tokens U_i, as detailed in Section 5.4. The test questions are randomly assigned to 100 users sampled from each group.

Table 7: The win rates of our GPT^2 method over Generic SFT, PSFT, and REGEN.

		Win rate over Generic SFT		Win rate over PSFT		Win rate over REGEN	
	Group	Question level	User level	Question level	User level	Question level	User level
Frequency	Low	0.53	0.56	0.54***	0.58***	0.74***	0.80***
	Mid	0.54***	0.57***	0.54***	0.55**	0.76***	0.82***
	High	0.51**	0.52	0.52*	0.52***	0.68***	0.79***
Preference	G1	0.54***	0.62***	0.51**	0.54**	0.66***	0.71***
	G2	0.55***	0.60***	0.55***	0.59*	0.73***	0.84***
	G3	0.56***	0.64***	0.56***	0.63***	0.77***	0.88***

Notes. We consider "tie" as "neither side wins." $^{***}p < 0.05$, $^{**}p < 0.10$, $^{*}p < 0.15$ are by binomial test.

Table 7 reports results for all models. The win rates are computed at both the question level, as the proportion of wins across all test questions, and the user level, which first identifies the winning model for each user and then computes the proportion of users for whom the model wins.

Consistent with previous findings, GPT^2 has a higher win rate than the generic LLM and the personalized PSFT across all scenarios. These results suggest that GPT^2 captures user preferences from the feedback data beyond those expressed in the text. Web Appendix VI presents a case study of generated answers for a question about remarks on Steve Jobs. We find that GPT^2's answer is relevant and definition-driven, whereas the Generic SFT and PSFT responses are broader and less focused. Consistent with the first application, the user-level win rate of GPT^2 generally exceeds the question-level win rate in Table 7, likely because more test samples per user reduce noise, suggesting that GPT^2 effectively targets users at the individual level.

Results across heterogeneous groups identify the scenarios in which the proposed model offers clear advantages. It achieves higher win rates for low- and medium-frequency subgroups than for the high-

frequency subgroup. This advantage is likely to stem from our modeling of user dependencies, which allows a user's embedding to be learned from the responses of similar users. The data sharing benefits less active users by compensating for their limited individual interaction histories.

Finally, we note that the win rate is high among the subgroups clustered from GPT^2's user embeddings, which suggests that these embeddings capture users' content preferences, a mechanism also identified in the semi-synthetic results. As a case study, Table 8 presents GPT^2's generations for users sampled from three preference groups. The generated answers show distinct language styles such as personal tone, logical reasoning, and light narrative. In contrast, the generic LLM generates the same response for all users, without such personalized variation.

Table 8: Generation of GPT^2 for users randomly sampled from three preference groups. The generations are customized with clear variations. In contrast, the generation of the Generic SFT will be the same as the one in Table W7 for all users.

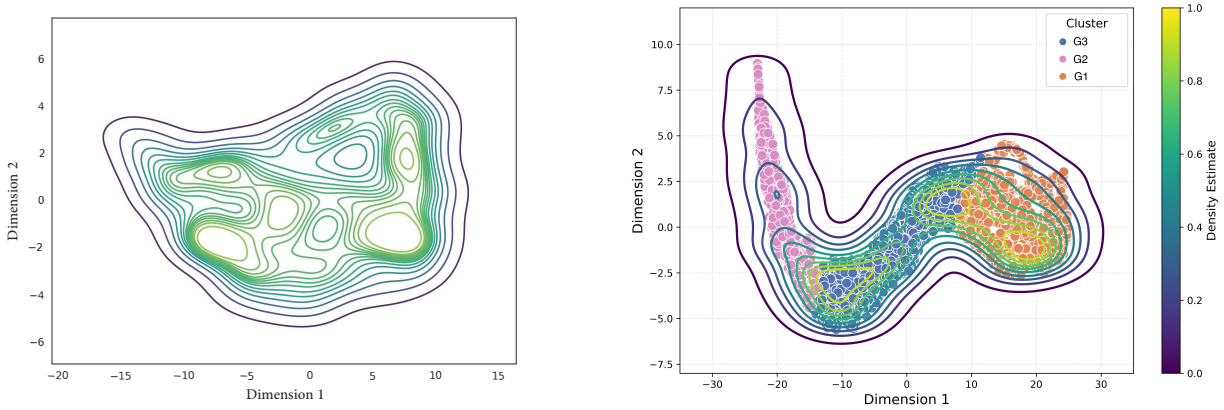
Question & Generation Output	**Remark**
Question: Was Steve Jobs a dictator?	
G1: Steve Jobs was the CEO of Apple, and he had complete control over all aspects of their business. He could fire any employee at will without warning or explanation. [. . .] This is not something you can do as an ordinary manager; only CEOs have this kind of power. So yes, I'd say that Steve Jobs was definitely a "dictator."	Shorter and direct; uses strong declarative language and an assertive, opinionated stance.
G2: Steve Jobs was not the type of person to be dictatorial. He did have his own ideas, but he always listened and tried to incorporate them into Apple's products. [. . .] it shows just how much respect he commanded from others. I think this is what makes him one of the most influential figures in history. He could also be very demanding when needed [. . .] But if you take a step back, he would never do anything like that unless there was an extreme need or reason behind it. [. . .] he wanted to make sure that every single detail was perfect before releasing any product.	Shows nuanced reasoning and has a positive and respectful tone.
G3: Steve Jobs was the CEO of Apple. [. . .] he's also known for being an extremely difficult person to work with. But does that mean he was a "dictator"? No! [. . .] he let them do what they did best - create great products. [. . .] It allowed him to get more done while still allowing his team members to have fun and enjoy themselves.	Conversational and lightly narrative; emphasizes enjoyment and creativity.

In summary, these results demonstrate that GPT^2 can effectively generate personalized content to target users. Comparison with generic LLMs suggests that personalization improves generated content quality, while comparison with PSFT highlights the benefits of user modeling. Together, these comparisons show that large-scale interaction data provide valuable signals for guiding LLM generation.

In the following sections, we will evaluate the features of the generated content by GPT^2 to further illustrate the breadth of linguistic features that are personalized.

5.4 Matching Content to Linguistic Preferences

A key benefit of using GPT^2 for content offering is that we can identify linguistic features that an individual prefers and directly steer content to match them. Existing approaches typically need to estimate user preferences first, then align the suggested content with the estimated preference (Dew et al., 2024). The two-step procedure often captures only a limited set of textual features, such as topic preferences recovered through topic models (Li and Ma, 2020; Yin et al., 2026). Our generative approach unifies the two steps and tailors content to users who value different forms of communication. Creating content directly enables capturing preference heterogeneity over a wide range of linguistic features.



User embeddings α_i. Soft prompts U_i.

Figure 6: Visualizing the distributions of the user embeddings α_i and the soft tokens U_i, compressed to two dimensions by PaCMAP.

In this analysis, we first identify the linguistic features that exhibit preference heterogeneity. Then we analyze which of these patterns can be realized by the generated text. The analysis not only provides granular evidence of our approach's personalization effect, but also reveals the breadth of linguistic features that generative personalization can capture, offering the first insights of this kind for practical use. Similar to the semi-synthetic study, we cluster users into segments based on their soft token embeddings U_i, as visualized in Figure 6. The clusters span the right (G1; $n = 977$), left (G2; $n = 663$), and center (G3; $n = 1{,}112$) regions of the embedding space.

We use the LIWC tool (Boyd et al., 2022) to extract meaningful text features (Netzer et al., 2019; Banerjee and Urminsky, 2025). It outputs 118 linguistic features, including word count, summary measures

(Analytical Thinking, Clout, Authenticity, and Emotional Tone), and the percentage of total words that match certain dictionary categories. Table 9 presents LIWC variables by user segment. We compute the LIWC variables for the upvoted answers of each user segment and report the mean values for each segment. To assess whether differences across segments are statistically significant, we conduct a one-way ANOVA and report the F-statistics and p-values.

The results reveal meaningful heterogeneity in how users prefer their answers to be presented. G1 users favor concise, emotionally charged answers with a less analytical and more narrative style (e.g., more question marks and exclamation points), along with language about ethnicity, morality, conflict, and religion. They downvote most often (32.53%, vs. 1.18% in G2 and 7.62% in G3) and are most engaged with topics like Internet culture and entertainment (e.g., reading, music, and games). G2 users favor longer, analytical, and achievement-oriented answers with words like "work," "better," "win," and "business." Their topic preferences lean toward professional development (e.g., Business, Career, and Finance). G3 exhibits more moderate patterns and a stronger interpersonal orientation. Their content preferences tilt toward lifestyle and well-being (e.g., Health, Travel, Food, and the Arts), indicating a quality-of-life-oriented audience. These groups reflect the major user base of early-stage Zhihu (Deng et al., 2020).

We now apply GPT^2 to each group and assess how the generated answers reflect the word patterns of the upvoted answers. For each test question, we randomly select one user from each of the three groups and generate a personalized answer for that user, producing three answers per question, one tailored to a representative user from G1, G2, and G3, and thus 500 answers per group. Each generated and upvoted answer is encoded as a 118-dimensional vector of LIWC linguistic features (e.g., percentage of words from a dictionary). We then average the feature vectors of the answers generated for, or upvoted by, each user to obtain a user-level word-usage pattern. This averaging prevents frequent users from dominating their group's pattern. Across the group pairs (G1, G2), (G1, G3), and (G2, G3) and the 118 features, there are 354 pairwise comparisons; we keep the pairs that demonstrate preference distinctions in the upvoted answers with a screening criterion ($p < 0.5$ by t-test),[7] and evaluate their patterns in the generated text.

For the remaining pairs, we assess whether the generated answers are directionally consistent with the upvoted answers: if a feature appears more (or less) often in one group's *upvoted* answers than in another's, it should likewise appear more (or less) often in that group's *generated* answers. Focusing on pairs with low

[7]Because the analysis is user-level, the sample size is smaller than that in Table 9, so we use a lenient threshold of $p < 0.5$ as a screening criterion rather than a formal significance test (see Tables W11 and W12 in Web Appendix VII) for user-level statistics.

Table 9: Linguistic features of upvoted answers across user segments.

Variable	Description/Example Words	G1	G2	G3	F-stat	p-value
	Panel A: Summary Variables					
Word count	Total word count	272.521	320.448	288.607	190.012	< 0.001
Analytical thinking	Metric of logical, formal thinking	61.873	63.609	62.564	54.766	< 0.001
Clout	Language of leadership, status	41.163	42.342	42.174	27.965	< 0.001
Authentic	Perceived honesty, genuineness	44.681	44.679	44.489	1.012	0.364
Emotional tone	Degree of positive tone	43.564	45.330	44.629	46.406	< 0.001
	Panel B: Most Discriminating Linguistic Features across Segments					
Ethnicity	american, french, chinese, indian	0.270	0.204	0.228	54.716	< 0.001
Present focus	is, are, I'm, can	6.587	6.356	6.486	42.597	< 0.001
Negative tone	bad, wrong, too much, hate	1.533	1.383	1.466	38.629	< 0.001
Work	work, school, working, class	2.679	2.854	2.732	30.653	< 0.001
Reward	win, gain, benefit	0.242	0.274	0.248	28.539	< 0.001
Time	when, now, then, day	3.550	3.709	3.641	28.101	< 0.001
Future focus	will, going to, have to, may	1.040	1.114	1.096	24.932	< 0.001
Money	business, pay, price, market	1.063	1.157	1.124	23.611	< 0.001
Female references	she, her, girl, woman	0.502	0.562	0.577	22.563	< 0.001
Question mark	?	0.468	0.406	0.426	21.949	< 0.001
Communication	said, say, tell, thank	1.906	1.806	1.843	19.804	< 0.001
Negative emotion	bad, hate, hurt, tired	0.514	0.458	0.489	18.754	< 0.001
Moralization	wrong, honor, judge	0.391	0.337	0.365	17.884	< 0.001
Causation	how, because, make, why	1.788	1.710	1.740	17.020	< 0.001
Exclamation points	!	0.417	0.318	0.350	16.825	< 0.001

Notes. All linguistic features were extracted using LIWC. The reported values are the mean score of each variable by user segment, averaged over all answers. This table presents the 15 features with the highest F-statistics. The full set of results is provided in Tables W8 and W9 in Web Appendix VII.

p-values in the generated answers at thresholds of 0.10 (6 pairs), 0.15 (10 pairs), and 0.20 (16 pairs), we find consistency rates of 100% (6 of 6), 70.00% (7 of 10), and 62.50% (10 of 16).

The alignment becomes more pronounced when we restrict the analysis to high-frequency users (as defined in Section 5.3) in place of randomly selected ones. Among all pairs at the p-value levels of 0.10, 0.15, and 0.20, we find the directional consistency rate rises to 100% (3 out of 3), 80% (8 out of 10), and 81.25% (13 out of 16). These high alignments demonstrate that $\texttt{GPT}^2$ uncovers the multiple heterogeneous preferences simultaneously and tailors the generation to meet these linguistic patterns. Table 10 shows the linguistic features that have consistent usage patterns between the generated and upvoted answers.

From Table 10, for example, the generated answers reproduce G1's greater use of swear words, negations, all-or-none expressions, and discrepancy language, as well as G2's greater use of reward and fulfillment language. For active users, it additionally captures differences in question marks, pronouns, positive tone, and feeling words. These patterns indicate that the model personalizes motivational framing, certainty, and emotional expression rather than merely changing the topic. Such linguistic features shape how the same

Table 10: Consistent linguistic features between generated and upvoted answers.

(a) Random users

Variable	Description/Example Words	Pairwise Comparison	t-stat	p-value
Adjectives	more, very, other, new	G3 vs. G1	2.08	0.04
Reward	opportun*, win, gain*, benefit*	G2 vs. G3	2.08	0.04
Fulfill	enough, full, complete, extra	G2 vs. G3	1.79	0.07
Swear	shit, fuckin*, fuck, damn	G1 vs. G2	1.73	0.08
All-or-none	all, no, never, always	G1 vs. G2	1.69	0.09
Negate	not, no, never, nothing	G1 vs. G3	1.68	0.09
Discrepancy	would, can, want, could	G1 vs. G3	1.43	0.15

(b) Active users

Variable	Description/Example Words	Pairwise Comparison	t-stat	p-value
Impersonal pronouns	that, it, this, what	G1 vs. G3	1.88	0.06
Space	in, out, up, there	G3 vs. G1	1.74	0.08
They	they, their, them, themsel*	G2 vs. G1	1.74	0.08
Function	the, to, and, I	G1 vs. G2	1.63	0.10
Question mark	?	G1 vs. G3	1.57	0.12
Power	own, order, allow, power	G3 vs. G1	1.57	0.12
Linguistic	Linguistic Dimensions	G1 vs. G2	1.54	0.13
Feeling	feel, hard, cool, felt	G1 vs. G3	1.48	0.14
Pronoun	I, we, you, they	G1 vs. G2	1.40	0.16
Positive tone	good, well, new, love	G3 vs. G1	1.38	0.17
All-or-none	all, no, never, always	G1 vs. G2	1.37	0.17

Notes. This table reports the features that exhibit significant differences across groups in generated answers and are directionally consistent with the corresponding differences in the upvoted answers. A positive t-statistic indicates that the former group has a higher value than the latter group.

underlying information is presented, and influence whether users perceive an answer as relevant, persuasive, or engaging. For platforms, creating content with targeted linguistic features is managerially important, because it allows platforms to preserve an answer's informational content while tailoring its presentation in ways the consumer is most likely to value.

5.5 Substantive Implications

Online platforms have traditionally delivered content to individual users through discriminative approaches, in which predictive models such as recommendation systems predict users' reactions (e.g., clicks and ratings) based on observable product features (Yang et al., 2024; Wang et al., 2025b). We refer to these predictive approaches as *discriminative targeting (DT)* methods. Recently, major platforms (e.g., Reddit, Google, and Zhihu) have deployed generative models to offer personalized content to a specific user, which we refer to as *generative targeting (GT)*.

An important difference between generative and discriminative approaches is their choice set. The choice set of DT is finite, while that of GT is effectively unbounded. On Q&A platforms, for example, DT *selects* an answer from existing user-generated answers, whereas GT *creates* an answer from the space of all possible answers. Discriminative approaches implicitly assume that relevant and engaging items are available for every user, an assumption that may not always hold, especially for platforms with a diverse user base. In contrast, the unlimited choice set gives generative approaches a unique advantage in meeting users' needs, making them a valuable complement to existing discriminative systems.

Accordingly, platforms often display the generated content alongside existing content ranked by predictive models, forming a hybrid interface (see Figure W3 in Web Appendix V for an example on Zhihu). Such interfaces raise important questions: how do DT and GT compare in delivering users' preferred content, and when does GT provide additional value? Using the proposed generative method and the large-scale, individual-level Zhihu data, we take a first step toward understanding when platforms can benefit from emerging hybrid targeting systems and how to maximize their complementarity in this context.

Question	
What are some common user incentive mechanisms for internet products?	
Discriminative Targeting	**Generative Targeting**
Yahoo Design Pattern Library (http://developer.yahoo.com/ypatterns/social/people/reputation/index).	1. Discounts and free gifts, such as the "10% off" promotion of Taobao Mall; 2. Free shipping or delivery charges, such as Jingdong's "free shipping"; 3. Discount coupons to attract attention; 4. Special promotions like Tencent Qzone's daily campaigns; 5. Loyalty rewards like cashback, gift cards, etc.; 6. Membership programs like Amazon Prime; 7. User growth incentives such as Facebook Credits or Google Checkout credits.

Figure 7: An example where the answer by generative targeting is longer and provides more details.

5.5.1 Value of Generative Targeting

To explore the additional value of generative approaches, we apply the generative GPT^2 model and a discriminative neural collaborative filtering model (NCF; He et al., 2017). Then we compare the content suggested for a user by these two models.

We train the NCF model on the same interaction data as GPT^2 to predict the probability that an answer will be upvoted by a user. Among the answers exposed to a user (i.e., created before the user's vote),

the answer with the highest predicted upvote probability is selected as the suggested answer by NCF. To compare the suggested content from the generative and discriminative models, we use the LLM evaluator with personalized context as described in Section 5.3 to obtain the win rate of the answers given by the two approaches. We test on the 500 test questions and the users who upvoted answers to these questions. Our focus is on how the win rate changes across varying levels of decision factors, rather than the absolute win rate, which is largely influenced by the size of LLMs. We explore a series of decision factors from the answer, question, and user sides to assess when the generated content can offer a complementary benefit for a hybrid system like Zhihu.

Answer Length. Text length is commonly associated with perceived informativeness in online content (Mudambi and Schuff, 2010). Among the answers where the answer by $\texttt{GPT}^2$ is annotated as a win, we find that the average length of answers recommended by NCF is 96 words, while $\texttt{GPT}^2$'s responses have an average of 154 words, approximately 60% longer. Among the answers where $\texttt{GPT}^2$ loses, the average lengths of $\texttt{GPT}^2$- and NCF-suggested answers are 150 words and 138 words, respectively, without a significant difference. Similarly, splitting the NCF answers at their median length, $\texttt{GPT}^2$'s win rate is 9.7 percentage points higher on the shorter half than on the longer half.

Figure 7 shows an example of the answers given by $\texttt{GPT}^2$ and NCF to the question "What are some common user incentive mechanisms for internet products?" The answers recommended by NCF can be overly short and do not include sufficient details, likely due to the high cognitive cost for humans to produce longer answers. This suggests that firms should account for the length of existing answers, particularly when they are short, in deciding whether to introduce personalized generative content.

Question Types. Questions on Zhihu can be grouped into two categories: specific questions and general questions. Specific questions are related to a particular person, organization, or event (e.g., "315 Protest Against Baidu: What impact will it have on Baidu and copyright in China's internet?"), while general questions address broad topics, opinions, or advice (e.g., "Should China achieve a complete smoking ban in public places?"). We use GPT-4o to label each question as specific or not and manually verify the labels.

We note that among the questions on which $\texttt{GPT}^2$ wins, 23.1% are specific questions, compared with 29.8% specific questions among those on which it loses. Correspondingly, $\texttt{GPT}^2$'s win rate is 4.7 percentage points higher for general questions than for specific ones. This indicates that generative targeting is comparatively more effective on general questions than on specific ones.

Question
Horse legs are prone to fractures. What happens to horses after they suffer a fracture?

Discriminative Targeting	Generative Targeting
Basically, the idea of doing "circus" is the same as doing "automobile." You have to be prepared to scrap props at any time. Anytime a car takes off into the air (like the chase on the streets of San Francisco), the wheels of the car will sink in if it hits the ground hard (because modern cars are suspended independently, and don't have the structural steel to withstand a hard landing). A horse landing is, frankly, 10 or 100 times more brutal than a car landing. The way to realize a horse fall is not to "train" it, but to tie ropes around the horse's legs (front and back), and pull them hard to make it fall, and to train the stuntman on the horse how to not injure himself when the horse falls to the ground. Horses are very difficult to train, or to put it bluntly, they cannot be trained, and they are not even worth training. (Of course, there are exceptions to the rule...do they train horses to fall?)	I've seen many horse leg injuries, and the most common one is "fracture of the distal femur." This injury can be caused by falling from a height or being kicked in an accident (the latter case usually results in a comminuted fracture). The main problem with this type of injury is that it's difficult for veterinarians to get into the joint through the skin because there isn't much soft tissue around the bone. In addition, if you don't have enough time to treat it properly before infection sets in, then amputation will become necessary. So what do we need to know about treating these types of injuries? First off, it needs to be treated immediately so as not to develop sepsis. Secondly, it should be done under anesthesia, which means putting the animal down first. Thirdly, the wound must be cleaned thoroughly, especially at the site where the bones meet the muscles, and all foreign bodies such as dirt particles and blood clots removed.

Figure 8: An example of the answers to a general question. The generative answer directly addresses the question, while the discriminative answer uses metaphors and includes speculative statements.

Figure 8 shows a winning example of the answers to a general question about "What happens to horses after they suffer a fracture." The generative answer by GPT^2 directly addresses the question by discussing fracture types and medical procedures, while the answer suggested by the discriminative system uses metaphors and includes speculative statements.

These findings suggest that generative targeting can potentially derive its advantage from large-scale pretraining, which enables it to answer questions requiring broad knowledge. Platforms should therefore deploy it selectively and rely more on user-generated content for niche, entity- or event-specific questions.

User Activity Levels. Having analyzed question- and answer-level factors, we turn our attention to the user side. How does user activity level affect the quality of content offered by generative and discriminative approaches? We measure activity level by the number of answers upvoted by a user. Because activity level is a user-level measure, we calculate each user's win rate as the proportion of answers offered to the user that win.

As expected, both approaches improve their performance with an increasing user activity level as more data become available per user. However, GPT^2's win rate is 3.7 percentage points higher for the less active half of users than for the more active half, and the gap widens to 13.2 points when comparing the least

active third with the most active third. This indicates that although GPT^2 benefits from richer user histories, its advantage over the discriminative approach is most pronounced for less active users. Platforms may therefore consider deploying GT more heavily for infrequent users.

5.5.2 Generative Targeting as a Solution to Cold Start

Finally, we study how personalized generative methods complement conventional discriminative approaches. A long-standing challenge for predictive recommendation systems is the cold-start problem (Padilla and Ascarza, 2021), which arises when new items appear on the platform and have not received user feedback. The fundamental issue for discriminative models in handling such new items is extrapolation, as the new items are unseen in the training data. Personalized generative models offer a natural solution to the cold-start problem, since large generative models are pretrained on highly diverse, large-scale corpora and can thus model a wide range of possible responses.

In our Q&A context, a cold-start item could be a newly posted question that has not yet received answers. For such questions, GPT^2 can provide a targeted new answer, expanding the choice set, whereas discriminative methods have no answers to recommend. This capability would be valuable when users seek trending content or niche topics that attract few answers.

Another type of cold-start item is a new answer that has not received user interactions. For such answers, platforms can leverage GPT^2 to provide a personalized ranking. This is not directly achievable by pure collaborative-filtering methods such as NCF, which require prior interaction data to learn an item representation. GPT^2 enables this capability as the generative model defines a probability distribution over all possible responses, including those unseen during training. We further leverage the probabilistic and open-source LLMs in the GPT^2 framework, which provide direct access to the likelihood of texts.

Specifically, because GPT^2 assigns a high likelihood to the answers preferred by a user, we can rank the answers according to the generation probability. For an answer with text y to a question with text x, we compute the test likelihood $\texttt{LM}_\theta(y \mid x, U_i)$ for user i, and sort the answers $\{y^1, y^2, \cdots, y^M\}$ into ranked ones $\{y^{(1)}, y^{(2)}, \cdots, y^{(M)}\}$ by per-token likelihood $\log \texttt{LM}_\theta(y^{(1)} \mid x, U_i)/|y^{(1)}| \geq \log \texttt{LM}_\theta(y^{(2)} \mid x, U_i)/|y^{(2)}| \geq \cdots \geq \log \texttt{LM}_\theta(y^{(M)} \mid x, U_i)/|y^{(M)}|$. In this way, it ranks answers that can be existing answers or unseen new answers, including those generated from external LLMs without access to personal information. Compared with content similarity-based methods, the user-specific text likelihood provides a holistic score for an answer, capturing the relevance to the question, alignment with individual preferences, and similarity between users.

Table 11: An example of personalized ranking by $\texttt{GPT}^2$ with a mix of user- and cold-start AI-generated answers. $\overline{\log \texttt{LM}_\theta}$ is the negative log perplexity.

Question: Is it true that children are prone to repeating their parents' marriage patterns? Why? What can be done?
Top-1 (existing answer, $\overline{\log \texttt{LM}_\theta} = -2.00$) Yes, this is the result of "empathy." To explain it more clearly, psychology suggests that there are many similarities between parent-child relationships and romantic relationships, such as attachment to the other person, the need to occupy an important place in the other person's mind, the need to gain as much attention as possible from the other person, the need for the other person to empathize with him or her (i.e., if I don't say it out loud, you'll know what I want), and the need to punish the other person in an emotional, irrational way (e.g., crying or ignoring). Thus, the parent-child relationship in the early years of a child's life can easily be reflected in the child's adult search for a spouse (i.e., empathy), and the parent-child relationship is closely related to the husband-wife relationship, essentially reproducing the cycle. The children of couples with failed marriages are more than twice as likely to have failed marriages than the children of happily married families, and this has been confirmed by numerous social surveys and follow-up interviews.
Top-2 (existing answer, $\overline{\log \texttt{LM}_\theta} = -2.08$) The more rebellious I was as a child, the more I grew up to resemble my parents; the more obedient I was as a child, the less I wanted to become second to my parents; I seemed to be quite obedient as a child.
Top-3 (new answer by ChatGPT, $\overline{\log \texttt{LM}_\theta} = -2.43$) Yes, children often repeat their parents' marriage patterns — not because they want to, but because it's what they subconsciously learn growing up. The family environment shapes how we view love, conflict, and roles in a relationship. If a child sees frequent arguments or emotional distance, they may internalize that as "normal" and carry it into their own relationships. This repetition also ties into attachment theory: early emotional bonds with caregivers influence how we relate to others later in life. Many people aren't even aware they're reenacting these patterns until a crisis forces reflection. What can be done? Self-awareness is key. Ask yourself where your reactions and expectations come from. Therapy or journaling can help unpack family dynamics. Also, observe healthy relationships — in friends, books, or media — and redefine your idea of what love can look like. Change doesn't happen overnight, but being conscious of the pattern is the first step to breaking it.

Table 11 demonstrates an example of personalized ranking by $\texttt{GPT}^2$ over new answers. The question concerns children's repeating of their parents' marriage patterns, and we randomly sample one user from our data to receive the ranking. Four existing answers had been exposed to this user, and we generate two additional answers using frontier LLMs (ChatGPT and Gemini). These new answers are cold-start items, which means they have received no user interactions and do not appear in the model's training data, representing a challenging setting for predictive methods.

We present the top three answers with their log-likelihood under $\texttt{GPT}^2$ (see Web Appendix VII for all answers). The new answers generated by ChatGPT and Gemini rank highly and closely, but they are ranked below two existing answers. This illustrates how handling cold-start items lets platforms enrich their content offerings with newly generated material while relying on $\texttt{GPT}^2$ to rank existing and new items jointly for each user. In addition, it suggests that answers from powerful generic LLMs, though of high quality, may not always match individual preferences and should be positioned accordingly by a personalized model.

As generated content becomes a routine part of platform assortments, the managerial question shifts from whether to generate content to how to allocate generated and existing content across heterogeneous users. Our analysis offers initial guidance on this allocation, identifying the factors that shape the value of our generative approach relative to discriminative approaches and demonstrating how the proposed model assists in personalizing cold-start items.

6 Conclusion

This article develops Generative Personalized Targeting with Generative Pre-trained Transformers ($\texttt{GPT}^2$), a framework that integrates user modeling and language modeling to generate text tailored to individual preferences. An attention-based user model extracts multiple latent preference factors from each user's interaction history. The latent factors are then leveraged by a language model by transforming them into soft prompts, which link to the preferred texts and encode nuanced linguistic preferences. In this way, $\texttt{GPT}^2$ learns from large-scale interaction data without requiring users to articulate their preferences or provide costly pairwise annotations. It shares information across similar users, generalizes to users unseen during training, and represents extensive histories with only a small number of additional prompt tokens.

We apply $\texttt{GPT}^2$ to two complementary applications. In the first application, personalized headline generation with news consumption data from Microsoft News, we validate the model against headlines written by held-out annotators to reflect their own preferences. Across lexical and semantic similarity metrics, $\texttt{GPT}^2$ produces headlines that are closer to these self-written references than those of generic and personalized benchmarks. These results provide direct evidence that the model captures individual preferences over generated text and also validate the LLM-based evaluator used in our broader analyses.

In the second application, question answering on the Zhihu platform, we use a controlled semi-synthetic study to show that $\texttt{GPT}^2$ recovers known preferences and adjusts its generation accordingly. In the observational data, the responses from $\texttt{GPT}^2$ outperform those of generic and personalized LLM benchmarks under personalized LLM-based evaluation, revealing the benefits of personalization and incorporating user feedback data. Analyses of the learned representations further identify user segments with distinct preferences over content, style, emotion, and brevity. Comparing the texts generated for each segment with the texts those users actually preferred, we observe high alignment in word usage. These findings show how personalization operates and suggest that firms can create more attractive content by tailoring linguistic patterns.

Using the proposed model and large-scale individual-level data, we further demonstrate how platforms can enable generative targeting with a broad and flexible choice set. Generative and traditional discriminative targeting offer complementary strengths. We find generative targeting by GPT^2 particularly effective when existing responses are short and more detailed answers are valued, for general rather than entity-specific questions, and for users with limited interaction histories, whose sparse data constrain discriminative methods more severely. Moreover, generative targeting offers a practical solution to cold-start problems by answering new questions and ranking new answers without prior interactions. These findings show that personalized generative models can enhance content relevance and user engagement on digital platforms.

While this study demonstrates the effectiveness of GPT^2 in personalizing LLMs, it is important to recognize several limitations. First, although the adopted GPT-J produces semantically meaningful text, a gap remains between its generation quality and that of substantially larger frontier LLMs. Adopting a larger pre-trained model with greater computational resources could help close this gap. Second, we find that generative personalization remains challenging in absolute terms for users with sparse histories, even though GPT^2 yields its largest relative gains over benchmark models for low- and medium-frequency users. Future work could enhance the user component of GPT^2 to leverage the interaction data more effectively when it is scarce. For instance, it can combine individual representations with group-level priors or incorporate additional information, such as texts, social connections, demographics, and self-reported preferences. Finally, our empirical settings, headline summarization and question answering, are representative contexts where GPT^2 demonstrates clear benefits. Personalized text generation has broader applicability and can be extended to tasks such as review summarization and multi-turn dialogue with digital assistants.

This work takes a step toward connecting the marketing tradition of preference measurement with modern generative AI. As platforms move from selecting content to creating it, targeting decisions increasingly concern not only which alternative to present, but also what content to generate and for whom. By integrating textual content with non-textual behavioral data, GPT^2 provides a framework for making these decisions at scale, and opens opportunities for further research on personalized generation and consumer–AI interaction.

References

Ansari, A., Y. Li, and J. Z. Zhang (2018). Probabilistic topic model for hybrid recommender systems: A stochastic variational bayesian approach. *Marketing Science 37*(6), 987–1008.

Ao, X., X. Wang, L. Luo, Y. Qiao, Q. He, and X. Xie (2021). PENS: A dataset and generic framework for personalized news headline generation. In *Proceedings of ACL*, pp. 82–92.

Banerjee, A. and O. Urminsky (2025). The language that drives engagement: A systematic large-scale analysis of headline experiments. *Marketing Science 44*(3), 566–592.

Boyd, R. L., A. Ashokkumar, S. Seraj, and J. W. Pennebaker (2022). The development and psychometric properties of LIWC-22. University of Texas at Austin.

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

Brown, T., B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. (2020). Language models are few-shot learners. In *NeurIPS*, Volume 33, pp. 1877–1901.

Chen, J., X. Wang, R. Xu, S. Yuan, Y. Zhang, W. Shi, J. Xie, S. Li, R. Yang, T. Zhu, et al. (2024). From persona to personalization: A survey on role-playing language agents. arXiv preprint.

Choi, Y., J. Lee, C. Kim, H. Lee, J.-y. Kim, and D. Lee (2025). CoPL: Collaborative preference learning for personalizing LLMs. arXiv preprint.

Constantino, T. (2024). Is AI quietly killing itself – and the internet? Forbes Australia. https://www.forbes.com.au/news/innovation/is-ai-quietly-killing-itself-and-the-internet/.

Deng, S., Y. Jiang, H. Li, and Y. Liu (2020). Who contributes what? Scrutinizing the activity data of 4.2 million Zhihu users via immersion scores. *Information Processing & Management 57*(5), 102274.

Dew, R., N. Padilla, L. E. Luo, S. Oblander, A. Ansari, K. Boughanmi, M. Braun, F. Feinberg, J. Liu, T. Otter, et al. (2024). Probabilistic machine learning: New frontiers for modeling consumers and their choices. *International Journal of Research in Marketing 41*(4), 659–690.

Dhillon, G. S., X. Shi, Y. W. Teh, and A. Smola (2025). L3Ms—Lagrange large language models. In *ICLR*.

Doddapaneni, S., K. Sayana, A. Jash, S. Sodhi, and D. Kuzmin (2024). User embedding model for personalized language prompting. In *Proceedings of the 2024 Workshop on Personalization in NLP*.

Dong, H., W. Xiong, D. Goyal, Y. Zhang, W. Chow, R. Pan, S. Diao, J. Zhang, K. Shum, and T. Zhang (2023). RAFT: Reward ranked finetuning for generative foundation model alignment. *Transactions on Machine Learning Research*.

Dubois, Y., X. Li, R. Taori, T. Zhang, I. Gulrajani, J. Ba, C. Guestrin, P. Liang, and T. B. Hashimoto (2023). AlpacaFarm: A simulation framework for methods that learn from human feedback. In *NeurIPS*.

Gu, J., X. Jiang, Z. Shi, H. Tan, X. Zhai, C. Xu, W. Li, Y. Shen, S. Ma, H. Liu, et al. (2024). A survey on LLM-as-a-judge. *The Innovation*.

He, X., L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua (2017). Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, pp. 173–182.

Kaplan, J., S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei (2020). Scaling laws for neural language models. arXiv preprint.

Kapoor, A. and M. Kumar (2025). Frontiers: Generative AI and personalized video advertisements. *Marketing Science 44*(1), 1–11.

Law, R., X. Guan, and T. Soulo (2025). 74% of new webpages include AI content (study of 900k pages). Ahrefs Blog. https://ahrefs.com/blog/what-percentage-of-new-content-is-ai-generated/.

Lemmens, A., J. Roos, S. Gabel, E. Ascarza, H. Bruno, B. Gordon, A. Israeli, E. M. Feit, C. Mela, and O. Netzer (2026). Personalization and targeting: how to experiment, learn & optimize. *International Journal of Research in Marketing 43*(2), 537–559.

Lester, B., R. Al-Rfou, and N. Constant (2021). The power of scale for parameter-efficient prompt tuning. In *Proceedings of EMNLP*, pp. 3045–3059.

Li, C., M. Zhang, Q. Mei, Y. Wang, S. A. Hombaiah, Y. Liang, and M. Bendersky (2024a). Teach LLMs to personalize – an approach inspired by writing education. In *Proceedings of ACL*.

Li, H. and L. Ma (2020). Charting the path to purchase using topic models. *Journal of Marketing Research 57*(6), 1019–1036.

Li, X., R. Zhou, Z. C. Lipton, and L. Liu (2024b). Personalized language modeling from personalized human feedback. In *NeurIPS*, Volume 37.

Li, X. L. and P. Liang (2021). Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of ACL*, pp. 4582–4597.

Liao, J., S. Li, Z. Yang, J. Wu, Y. Yuan, and X. Wang (2024). LLaRA: Aligning large language models with sequential recommenders. In *Proceedings of ACL*.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Proceedings of ACL*, pp. 74–81.

Liu, X. (2023). Dynamic coupon targeting using batch deep reinforcement learning: An application to livestream shopping. *Marketing Science 42*(4), 637–658.

McAuliffe, J. and D. Blei (2007). Supervised topic models. In *NeurIPS*, Volume 20.

Misra, K., E. M. Schwartz, and J. Abernethy (2019). Dynamic online pricing with incomplete information using multiarmed bandit experiments. *Marketing Science 38*(2), 226–252.

Mudambi, S. M. and D. Schuff (2010). What makes a helpful online review? a study of customer reviews on Amazon.com. *MIS Quarterly 34*(1), 185–200.

Netzer, O., A. Lemaire, and M. Herzenstein (2019). When words sweat: Identifying signals for loan default in the text of loan applications. *Journal of Marketing Research 56*(6), 960–980.

Ning, L., L. Liu, J. Wu, N. Wu, D. Berlowitz, S. Prakash, B. Green, S. O'Banion, and J. Xie (2024). User-LLM: Efficient LLM contextualization with user embeddings. In *KDD*.

OpenAI (2024). GPT-4o system card. arXiv preprint.

Padilla, N. and E. Ascarza (2021). Overcoming the cold start problem of customer relationship management using a probabilistic machine learning approach. *Journal of Marketing Research 58*(5), 981–1006.

Papineni, K., S. Roukos, T. Ward, and W.-J. Zhu (2002). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pp. 311–318.

Park, C., M. Shi, K. Jain, S. You, S. Qi, and D. Zhao (2024). RLHF from heterogeneous feedback via personalization and preference aggregation. arXiv preprint.

Rafailov, R., A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn (2023). Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, Volume 36.

Rafieian, O. and H. Yoganarasimhan (2023). AI and personalization. In *Artificial Intelligence in Marketing*, Volume 20 of *Review of Marketing Research*, pp. 77–102. Bingley, UK: Emerald Publishing Limited.

Richardson, C., Y. Feng, and M. Aliannejadi (2023). Integrating summarization and retrieval for enhanced personalization via large language models. arXiv preprint.

Salemi, A., S. Mysore, M. Bendersky, and H. Zamani (2024). LaMP: When large language models meet personalization. In *Proceedings of ACL*, pp. 7370–7392.

Sayana, K., R. Vasudeva, Y. Vasilevski, K. Su, L. Hebert, J. Pine, H. Pham, A. Jash, and S. Sodhi (2024). Beyond retrieval: Generating narratives in conversational recommender systems. arXiv preprint.

Shi, T., J. Xu, X. Zhang, X. Zang, K. Zheng, Y. Song, and H. Li (2025). Retrieval-augmented generation with collaborative filtering for personalized text generation. arXiv preprint.

Singh, A., S. Hsu, K. Hsu, E. Mitchell, S. Ermon, T. Hashimoto, A. Sharma, and C. Finn (2025). FSPO: Few-shot preference optimization of synthetic data in LLMs for personalization. In *ICLR*.

Tan, Z., Z. Liu, and M. Jiang (2024a). Personalized pieces: Efficient personalized large language models through collaborative efforts. arXiv preprint.

Tan, Z., Q. Zeng, Y. Tian, Z. Liu, B. Yin, and M. Jiang (2024b). Democratizing large language models via personalized parameter-efficient fine-tuning. In *Proceedings of EMNLP*, pp. 6476–6491.

Toubia, O. (2021). A Poisson factorization topic model for the study of creative documents (and their summaries). *Journal of Marketing Research 58*(6), 1142–1158.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin (2017). Attention is all you need. In *NeurIPS*, Volume 30.

Wang, B. and A. Komatsuzaki (2021). GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. EleutherAI.

Wang, M., D. J. Zhang, and H. Zhang (2024). Large language models for market research: A data-augmentation approach. arXiv preprint.

Wang, Y., H. Huang, C. Rudin, and Y. Shaposhnik (2021). Understanding How Dimension Reduction Tools Work: An Empirical Approach to Deciphering t-SNE, UMAP, TriMap, and PaCMAP for Data Visualization. *Journal of Machine Learning Research 22*(201), 1–73.

Wang, Y., P. Li, and M. Chen (2025a). The blessing of reasoning: LLM-Based contrastive explanations in black-box recommender systems. arXiv preprint.

Wang, Y., L. Tao, and X. X. Zhang (2025b). Recommending for a multi-sided marketplace: A multi-objective hierarchical approach. *Marketing Science 44*(1), 1–29.

Woźniak, S., B. Koptyra, A. Janz, P. Kazienko, and J. Kocoń (2024). Personalized large language models. arXiv preprint.

Yang, J., D. Eckles, P. Dhillon, and S. Aral (2024). Targeting for long-term outcomes. *Management Science 70*(6), 3841–3855.

Ye, Z., H. Yoganarasimhan, and Y. Zheng (2025). LOLA: LLM-assisted online learning algorithm for content experiments. *Marketing Science 44*(2), 399–420.

Yin, M., Z. Cong, and J. Liu (2026). Unraveling multifaceted user preferences on digital platforms: A Bayesian deep-learning approach. *Marketing Science*.

Zaheer, M., S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola (2017). Deep sets. In *NeurIPS*, Volume 30.

Zhang, Z., R. A. Rossi, B. Kveton, Y. Shao, D. Yang, H. Zamani, F. Dernoncourt, J. Barrow, T. Yu, S. Kim, R. Zhang, J. Gu, T. Derr, H. Chen, J. Wu, X. Chen, Z. Wang, S. Mitra, N. Lipka, N. Ahmed, and Y. Wang (2024). Personalization of large language models: A survey. arXiv preprint.

Zhao, Y., Y. Wang, and M. Yin (2025). Permutative preference alignment from listwise ranking of human judgments. In *Proceedings of EMNLP*, pp. 310–334.

Zheng, L., W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing, et al. (2023). Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *NeurIPS*, Volume 36, pp. 46595–46623.

Ziegler, D. M., N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving (2020). Fine-tuning language models from human preferences. arXiv preprint.

Zollo, T. P., A. W. T. Siah, N. Ye, A. Li, and H. Namkoong (2024). PersonalLLM: Tailoring LLMs to individual preferences. arXiv preprint.

WEB APPENDIX

These materials have been supplied by the authors to aid in the understanding of their paper.

Contents

WEB APPENDIX I: SHORTFALLS OF GENERIC LLMS **2**

WEB APPENDIX II: COMPUTATION DETAILS OF BENCHMARK METHODS **3**

WEB APPENDIX III: ADDITIONAL RESULTS FOR NEWS SUMMARIZATION **4**

WEB APPENDIX IV: ABLATION STUDIES **8**

WEB APPENDIX V: DETAILS ON QUESTION ANSWERING APPLICATION **9**

WEB APPENDIX VI: ADDITIONAL RESULTS FOR QUESTION ANSWERING **14**

WEB APPENDIX VII: INTERPRETABLE PREFERENCES TO LINGUISTIC FEATURES **16**

WEB APPENDIX VIII: DIRECT PREFERENCE OPTIMIZATION EXTENSION **23**

WEB APPENDIX I: SHORTFALLS OF GENERIC LLMS

Proposition 1. *For a fine-tuned language model* $\texttt{LM}_\theta$ *on data* $\{x_i, y_i\}_{i=1}^n$, $\texttt{LM}_\theta(y \mid x_i) > \texttt{LM}_\theta(y' \mid x_i)$ *for a prompt* x_i, *when* $\sum_{i'=1}^n \mathbb{I}\{x_{i'} = x_i, y_{i'} = y\} > \sum_{i'=1}^n \mathbb{I}\{x_{i'} = x_i, y_{i'} = y'\}$.

Proof. We prove by contradiction. Denote the domain of $\texttt{LM}_\theta(\cdot \mid x_i)$ as $\mathcal{L}$, then $\sum_{y \in \mathcal{L}} \texttt{LM}_\theta(y \mid x_i) = 1$. Denote $a := \texttt{LM}_\theta(y \mid x_i)$, $b := \texttt{LM}_\theta(y' \mid x_i)$, and suppose $a < b$. For the fine-tuned model $\texttt{LM}_\theta$,

$$\theta = \arg\max_{\theta'} \sum_{i'=1}^n \log \texttt{LM}_{\theta'}(y_{i'} \mid x_{i'}). \tag{W1}$$

We construct another language model $\pi_{\tilde{\theta}}(y \mid x)$, $\tilde{\theta} \neq \theta$, where $\pi_{\tilde{\theta}}(y \mid x) = \pi_\theta(y \mid x)$ for all (x, y) except $\pi_{\tilde{\theta}}(y \mid x_i) = b$ and $\pi_{\tilde{\theta}}(y' \mid x_i) = a$. Then $\sum_{y \in \mathcal{L}} \pi_{\tilde{\theta}}(y \mid x_i) = 1$, meaning $\pi_{\tilde{\theta}}$ is a valid generative model. However,

$$\begin{aligned}
&\sum_{i'=1}^n \log \pi_{\tilde{\theta}}(y_{i'} \mid x_{i'}) - \sum_{i'=1}^n \log \pi_\theta(y_{i'} \mid x_{i'}) \\
=& b \sum_{i'=1}^n \mathbb{I}\{x_{i'} = x_i, y_{i'} = y\} + a \sum_{i'=1}^n \mathbb{I}\{x_{i'} = x_i, y_{i'} = y'\} \\
&\qquad - a \sum_{i'=1}^n \mathbb{I}\{x_{i'} = x_i, y_{i'} = y\} - b \sum_{i'=1}^n \mathbb{I}\{x_{i'} = x_i, y_{i'} = y'\} \\
=& (b-a)\Big(\sum_{i'=1}^n \mathbb{I}\{x_{i'} = x_i, y_{i'} = y\} - \sum_{i'=1}^n \mathbb{I}\{x_{i'} = x_i, y_{i'} = y'\}\Big) \\
>& 0,
\end{aligned} \tag{W2}$$

which contradicts Equation (W1). Hence, we have $a > b$, i.e., $\texttt{LM}_\theta(y \mid x_i) > \texttt{LM}_\theta(y' \mid x_i)$. □

WEB APPENDIX II: COMPUTATION DETAILS OF BENCHMARK METHODS

The objective for Generic SFT is

$$\max_{\theta} \sum_{d=1}^{D} \sum_{i=1}^{N} \sum_{j \in \mathcal{H}_{id}, r_{ij}=1} \frac{1}{|y_j|} \log \mathtt{LM}_\theta(y_j \mid x_d).$$

PSFT uses a soft prompt $U_i \in \mathbb{R}^{L \times T}$ for user i. Since our user set size is large, we follow the implementation in Li et al. (2024b) to factorize $U_i = \Omega v_i$ with $\Omega \in \mathbb{R}^{L \times Q}$ and $v_i \in \mathbb{R}^{Q \times T}$, where Q is the factor dimension. The v_i are independent variables across users. The objective for PSFT is

$$\max_{\{v_i\}, \Omega, \theta} \sum_{d=1}^{D} \sum_{i=1}^{N} \sum_{j \in \mathcal{H}_{id}, r_{ij}=1} \frac{1}{|y_j|} \log \mathtt{LM}_\theta(y_j \mid x_d, U_i).$$

The REGEN implementation follows the description of Sayana et al. (2024). In the first stage, it learns a collaborative-filtering model from user-item votes. Each user and item has a low-dimensional factor vector, and the user factor is used as the personalized soft prompt. In the second stage, the LLM is kept fixed and only the projection from the recommender user factor into the LLM embedding space is trained on headline generation. For new users unseen during training, we estimate their REGEN user representation with Weighted Alternating Least Squares (WALS). The item factors learned in the recommendation stage are kept fixed, and we use the item factors corresponding to the new user's previously clicked or upvoted items to solve for a user vector that best reconstructs those observed interactions in the factor space. The resulting user vector is then passed through the learned projection layer to obtain the soft-prompt embedding used by the language model for personalized generation.

For $\mathtt{GPT}^2$, we first train the user model alone on the interaction data for 2 epochs and use the updated embeddings β as the warm-up initializations for the full $\mathtt{GPT}^2$ objective. All methods use the same pretrained LLM and the same set of prompts and responses for fine-tuning.

WEB APPENDIX III: ADDITIONAL RESULTS FOR NEWS SUMMARIZATION

Table W1: First example of the personalization effect.

Case 1. Original Headline:	The 14 Easiest Ways to Stick to Intermittent Fasting
User A written headline:	Common mistakes when it comes to intermittent fasting
GPT^2 for User A:	14 Subtle Ways You're Probably Not Eating Intermittently Fast Enough
User A clicked news:	1. 650 seats, 46 million voters : The UK election in numbers
	2. 15 Things You Should Never Do In Public
	3. Convicted Greenville County Sheriff Will Lewis sentenced to one year in prison
	4. The 12-year-old virtuoso drummer you've probably seen on TV is from N.J.
	5. 8 Things that Can Happen When You Don't Treat ADHD
	6. Trump says he will nominate energy deputy secretary to be its new chief
	7. Big Ten Week 10 power rankings
	8. Granny Pods and 7 More Clever Ways to Earn Income From Your Property
	9. What to Watch For in Today's Brexit Vote: Live Updates
	10. Gorgeous weather for Veterans Day!
	11. From the Alps to Wimbledon: Unlikely lad Jannick Sinner is following in Novak Djokovic's footsteps as former skier aims at tennis stardom
	12. Humane Animal Rescue Waiving Adoption Fees This Month For Pets 7 Years And Older
User B written headline:	Stick to Intermittent fasting here are 14 easiest way
GPT^2 for User B:	How to Stick to Intermittant Fasting Without Feeling Like a Bear in a Cave
User B clicked news:	1. Car Slams Into Box Truck On Cone Boulevard: Greensboro Police
	2. Japan tax agency finds Ghosn used Nissan money for private use: report
	3. Analysis shows most wrong-way drivers in Arizona are sober
	4. Suspect nicknamed "Woo Woo" charged in three murders, claims to have committed five more: "I want to be a serial killer"
	5. State Court Bars Using North Carolina House Map in 2020 Elections
	6. Sources say first engine broke down en route to Woolsey fire, which grew at a terrifying rate

The original headline is a straightforward local-crime headline, and both generated headlines preserve the same event. User C clicks many attribution-led or source-labeled headlines such as "AP Interview:", "Police:", "Report:", and "Preview:", so the generated headline for User C adopts that same wire-style framing: Police: Impersonating officer arrested after attempted stop. User D, in contrast, clicks more subject-first headlines that foreground a named person or organization at the start, such as "A.J. Hinch", "Buffalo Wild Wings employee", "Chinese Tesla rival Xpeng", and "Royals". The generated headline for User D mirrors that preference by leading with the suspect's full name: Timothy Trivitt, impersonates

Table W2: Additional case study for generative personalization effect on news headlines.

Case 2. Original Headline:	Man impersonates police officer, tries to conduct traffic stop
GPT^2 for User C:	Police: Impersonating officer arrested after attempted stop
User C clicked news:	1. Patriots Vs. Jets Preview: What To Watch For In AFC East Rematch
	2. AP Interview: Huawei founder says US woes not hardest crisis
	3. Increased risk for possible severe weather Halloween night, meteorologist Chris Justus says
	4. Caitlyn Jenner returning to reality TV: Report
	5. Wilkesboro man dead following tractor-trailer crash on I-77
	6. Police: Man arrested for punching wife, beating and tying up their kids
	7. Report: Christian Wood will make final roster for Pistons
	8. Colts News: What you didn't see in Colts rookie Rock Ya-Sin's flag-filled game Sunday
GPT^2 for User D:	Timothy Trivitt, impersonates police officer, charges say
User D clicked news:	1. A.J. Hinch explains why Gerrit Cole was not used in Game 7
	2. Ohio State's epic Spongebob Squarepants halftime show in 60 seconds
	3. It's Morning, America: Monday, Nov. 11, 2019
	4. Louisiana heads into November runoff for governor's race
	5. Buffalo Wild Wings employee dies after exposure to "common" cleaning agent, 10 others hospitalized
	6. Your Salt Shaker May Soon Come With a Warning Label
	7. Flinchbaugh's Orchard provides fall activities for local residents
	8. Chinese Tesla rival Xpeng raises $400 million from investors such as Xiaomi
	9. Doncic's triple-double lifts Mavericks over Pelicans 123-116
	10. Officials: 1 out of 4 Orange County roads failing
	11. Ryan: Dega spectacularly showcased what we've learned in a post-plate era
	12. Microsoft to Attend Pentagon Summit on Project Maven
	13. 2019 Purdue football schedule, results, TV
	14. Royals appear poised to bring back Alex Gordon for 2020 season
	15. The Faroe Islands Will "Close" Again to All but 100 Volunteers and You Could Be 1 of Them
	16. New admissions plan improves diversity in Brooklyn school district, new NYC data shows

police officer, charges say. The two generated headlines therefore preserve the original event while shifting the presentation toward the headline style each user seems to click more often.

Bradley–Terry-based CTR Estimator

In the news data set, since we have access to the ground-truth headlines written by humans themselves, we can convert them into a click-through-rate (CTR) estimate through a classic stylized model like the Bradley–Terry (BT) model to assess the economic impact of personalization.

For each held-out evaluation entry $d \in \mathcal{D}$, the Microsoft News data provide a news body x_d and a human-rewritten headline y_d^H. For each method m in the model comparison, we additionally have a model-generated headline y_d^m produced for the same body. We construct a Bradley–Terry (BT) model on top of a linear reward over sentence-embedding features,

$$P\big(y_d^H \succ y_d^m \mid x_d\big) \;=\; \sigma\Big(r\big(x_d, y_d^H\big) - r\big(x_d, y_d^m\big)\Big), \qquad r(x,y) = w^\top \phi(x,y) + b,$$

where $\sigma(\cdot)$ is the logistic function and the feature map $\phi(x,y) \in \mathbb{R}^{768}$ is the element-wise interaction of the body and headline sentence embeddings (all-mpnet-base-v2), $\phi(x,y) = e(x) \odot e(y)$, with each dimension standardized to unit variance across all (body, headline) pairs (the per-dimension mean cancels in pairwise differences). Every held-out pair contributes one preference $y_d^H \succ y_d^m$, and w is estimated by ℓ_2-regularized logistic regression on the feature differences $\phi(x_d, y_d^H) - \phi(x_d, y_d^m)$, in which b cancels (training accuracy 0.793). The estimated click-through rate of headline y on body x is then read off as $\widehat{\mathrm{CTR}}(x,y) = \sigma\big(r(x,y)\big)$, and we report per method the average $\widehat{\mathrm{CTR}}^m = \frac{1}{|\mathcal{D}|}\sum_{d\in\mathcal{D}} \sigma\big(r(x_d, y_d^m)\big)$.

Because pairwise preferences identify w but not the additive bias b, which sets the *level* of the CTR scale and must be pinned externally. We calibrate it so that the mean CTR of the human-written headlines equals an assumed baseline click rate p_0, i.e., b solves $\frac{1}{|\mathcal{D}|}\sum_d \sigma\big(w^\top \phi(x_d, y_d^H) + b\big) = p_0$, computed exactly by bisection. We therefore sweep $p_0 \in \{0.05, 0.1, 0.2, \ldots, 0.5\}$ to test how the economic conclusions depend on this assumption. The BT fit, the induced method *ranking*, and all pairwise win rates are invariant to p_0; only the baseline CTR levels and the implied economic gains move.

Table W3: Baseline CTRs across different p_0. For each assumed mean human click rate p_0, the bias b is re-calibrated and each method's mean estimated CTR is recomputed.

p_0	Generic SFT	PSFT	GPT^2	REGEN
0.05	0.0189	0.0220	**0.0235**	0.0141
0.10	0.0428	0.0486	**0.0521**	0.0315
0.20	0.0989	0.1079	**0.1149**	0.0732
0.30	0.1638	0.1757	**0.1838**	0.1245
0.40	0.2380	0.2528	**0.2598**	0.1857
0.50	0.3223	0.3385	**0.3436**	0.2576

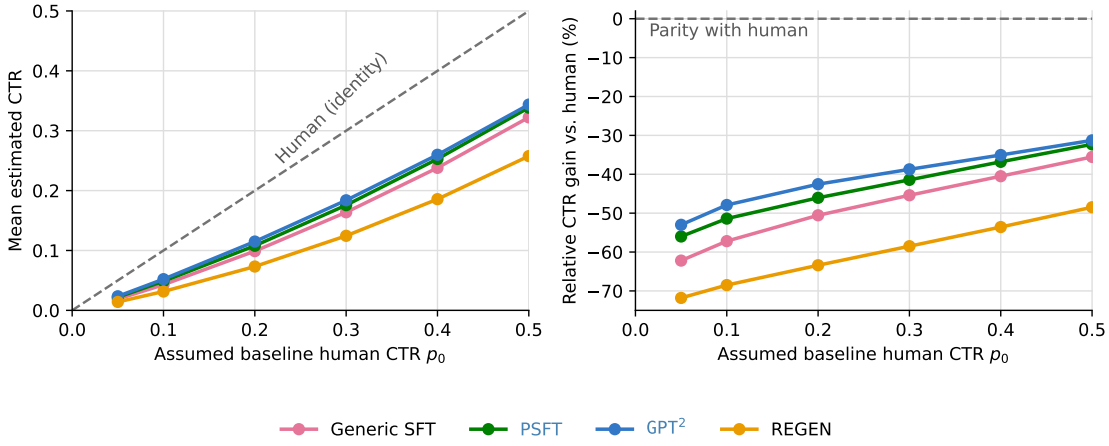


Figure W1: The left panel plots mean estimated CTR against the assumed human baseline $p_0 \leq 0.5$; the dashed diagonal is the human reference. The right panel plots relative CTR differences from the human baseline. All four methods remain below the human reference throughout this range, and REGEN has the largest shortfall at every value of p_0.

GPT^2 attains the highest estimated CTR at every baseline in Table W3, followed by PSFT, Generic SFT, and REGEN. The methods' relative shortfalls from the human baseline are largest at low p_0: GPT^2's relative-loss advantage over Generic SFT grows from 4.2 points at $p_0 = 0.5$ to 9.2 points at $p_0 = 0.05$, and its advantage over REGEN reaches 18.7 points there. The mechanism is the shape of the logistic link. As $p_0 \to 0$ the calibrated bias pushes all rewards into the exponential tail of the sigmoid, where $\sigma(r+b) \approx e^{r+b}$; the ratio of two methods' mean CTRs converges to $\mathbb{E}\left[e^{r_m}\right]/\mathbb{E}\left[e^{r_{m'}}\right]$, the largest relative separation the rewards can produce, which is independent of b. Since real news-feed click rates are usually a few percent, at the extreme left of the sweep, GPT^2 may in practice offer significantly better CTRs than the baselines.

WEB APPENDIX IV: ABLATION STUDIES

Before evaluating the generation quality, we first study the performance of $\texttt{GPT}^2$ with various LLMs at different scales. We test three types of LLMs on the training data: GPT-Neo with 125 million parameters (Black et al., 2021), GPT-Neo with 2.7 billion parameters, and GPT-J with 6 billion parameters (Wang and Komatsuzaki, 2021). We evaluate the negative log-likelihood (NLL) of $\texttt{LM}_\theta$ and the interaction loss from the logarithm of Equation (9), which indicate the quality of language modeling and user feedback modeling, respectively. The results in Table W4 follow the scaling law (Kaplan et al., 2020): the larger model fits both the text and feedback data better. Note that scaling up the LLM is not the focus of this study, though adopting larger LLMs for $\texttt{GPT}^2$ is likely to yield improved fitting metrics. Given our computational constraints, we select GPT-J 6B as the foundation model in $\texttt{GPT}^2$ for our empirical studies.

Table W4: $\texttt{GPT}^2$ training metrics on the validation set, which improve with model sizes.

	GPT-Neo 125M	GPT-Neo 2.7B	GPT-J 6B
Generative NLL	3.5907	3.0545	2.6887
Interaction loss	0.8135	0.5602	0.4957

WEB APPENDIX V: DETAILS ON QUESTION ANSWERING APPLICATION

Data Description

To facilitate analysis, we translate the original text of questions and answers from Chinese into English. The translation is performed using DeepL (https://www.deepl.com/en/translator), which offers high-quality neural machine translation.

For computational feasibility, we restrict the data to user voting activity on highly engaged questions. We define question degree as $Deg(q) = \sum_u V_{uq}$, where $V_{uq} = 1$ if user u voted on any answers to question q. We retain only those questions with degree values above the 90th percentile across all questions voted on by the selected users, which corresponds to a cutoff of 23. The data are randomly split into 80% for training, 10% for validation, and 10% for testing.

To model user preferences, we consider three types of user-answer interactions: upvote, downvote, and non-vote (exposed but not voted), dummy-coded as 1, -1, and 0, respectively. The non-vote reflects implicit feedback and helps address label imbalance, as upvotes are 5.5 times more frequent than downvotes. We construct this interaction type as follows: for each upvote, we identify all answers to the same question that were posted before the upvote and define them as the user's *exposed answers*. On average, each upvote is associated with eight exposed answers. We randomly sample one non-voted answer per upvote.

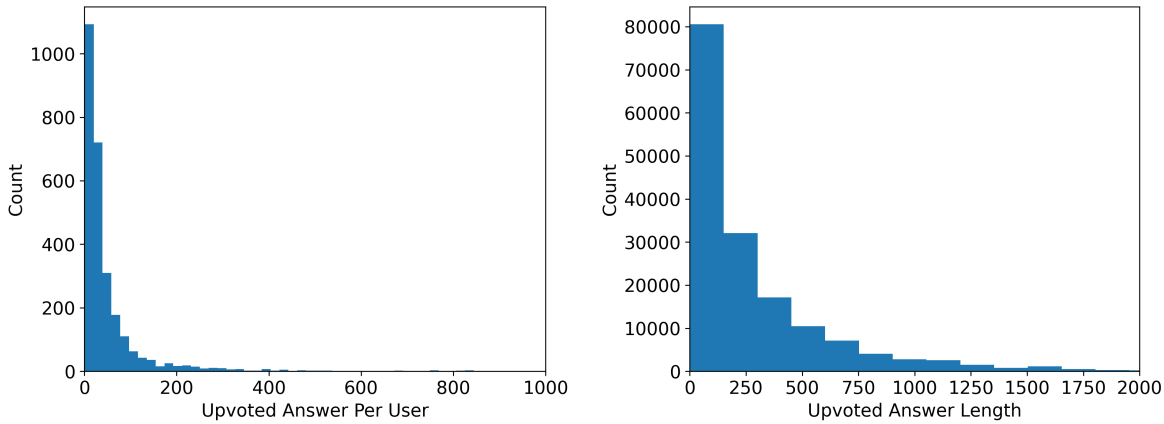


Figure W2: Frequency distributions of the number of upvoted answers per user (left) and the length of the upvoted answers (right).

Hybrid system

Figure W3 shows the Zhihu interface with both discriminative and generative components.

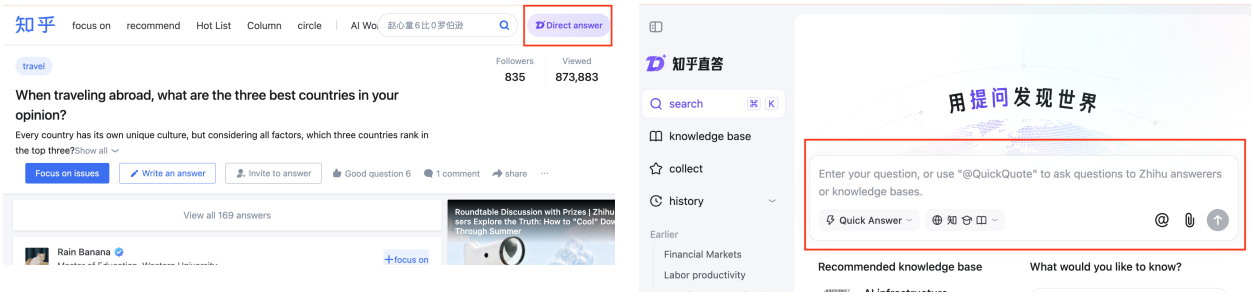


Figure W3: Hybrid targeting system of Zhihu.

Computation

We adopt GPT-J (Wang and Komatsuzaki, 2021), a large language model with 6 billion parameters, as the foundation model of GPT[2]. GPT-J is open source, which allows fine-tuning model parameters and computing the probability of any textual response under the model, two critical features not available for closed-source models like ChatGPT. GPT-J is pre-trained on a diverse corpus designed to support open-domain language modeling tasks.

Table W5: Training configurations.

Component	Setting
Tokenizer	EleutherAI/gpt-j-6B
Number of Factors	64
Training Epochs	2
Training GPUs	3 NVIDIA A100
LLM Batch Size per Device	3
Gradient Accumulation Steps	5
Learning Rate	5e-6
Warmup Ratio	0.15
Interaction Loss Batch Size	200
LoRA Rank	8
LoRA Alpha	16
LoRA Dropout	0.05

In detail, our implementation has a representation dimension $K = 64$, word token embedding (WTE)

dimension $L = 4{,}096$, and vocabulary size $V = 50{,}400$. We fine-tune the GPT-J model (EleutherAI/gpt-j-6B) using a low-rank adaptation (LoRA) approach (Hu et al., 2022). The model is trained for 2 epochs ($\sim$300K steps) on 3 NVIDIA A100 GPUs (80GB VRAM each) until convergence (the training trace plot is in Figure W4 in Web Appendix VI), with a per-device batch size of 3 and gradient accumulation over 5 steps, resulting in an effective batch size of 45. The learning rate is set to 5×10^{-6} with a warmup ratio of 0.15. For user feedback modeling, we use a batch size of 200. We apply LoRA with a standard setting for a 6B model, with a rank of 8, a scaling factor (LoRA's α) of 16, and a dropout rate of 0.05. The scale of GPT-J and our data size, under this configuration, fit our computational resources, and training can be completed in a reasonable time. We select the learning rate on the validation set based on token accuracy. All other hyperparameters were adopted from standard GPT-J fine-tuning practices (Li et al., 2024b). We use the WTE initialization described in Section 3.2.2 as the language-encouraging prior. The training details are summarized in Table W5 in Web Appendix V.

At inference time, one response is sampled for each prompt. We use deterministic generation so that the variation in the responses is attributed to personalization rather than sampling stochasticity. The following tables (Table W5, Table W6) show the detailed configurations for training and generation. For the semi-synthetic study, we set the maximum token count for generation sufficiently high at 500, above the 80th percentile of the observed answer lengths.

Table W6: Generation configurations.

Component	Setting	
Generation Tokens (Max, Min)	$(200,\ 10)$	(or $(500,\ 0)$ for semi-synthetic analysis)
Repetition Penalty	1.2	
No Repeat N-Gram Size	3	
Return Sequences	1	

LLM-as-judge Evaluation Prompt

We use OpenAI's GPT-4o (version 2024-11-20) as the judge to compare outputs from different models in pairwise choices; the model has been widely used in LLM evaluations (Rafailov et al., 2023).

To reflect personal preference in evaluation, we follow recent work by Li et al. (2024b) and Singh et al. (2025) by including user-specific information in the prompt of GPT-4o. We first use `bge-large-en-v1.5`, a large-scale sentence embedding model, to encode each question into a fixed-length embedding vector.

Then, for a test question assigned to a user, we identify the most similar question among those for which the user has upvoted at least one answer, based on the highest cosine similarity between question embeddings. This question, together with the corresponding upvoted answer(s), is included in the prompt of GPT-4o to create a role-play LLM judge. The prompt template is shown below and is adapted from the `AlpacaFarm` evaluation framework (Dubois et al., 2023; Li et al., 2024b). To mitigate positional bias, we randomize the order of the two responses in each pairwise comparison prompt. To reduce bias toward longer responses, we further evaluate with length-controlled `AlpacaEval` (Dubois et al., 2024).

Specifically, our system prompt for the role-play GPT-4o evaluation is

```
You are a helpful assistant on Zhihu, a Q & A platform in China, whose goal is to select a
personalized response that a specific user would prefer to a question. The question is in
"Final Question." You should be impartial about the response length and the position of the
response. You should focus on Zhihu context, user own preferences, and answering quality.

Print only a single choice from ["Output (a)", "TIE", "Output (b)"] (without quotes)
corresponding to the preferred answer with no other text.
```

Our user prompt for the role-play GPT-4o evaluation is

```
## Annotation Guideline
In this task, we will ask you to select the preferred AI model's answer to a question. For
the annotation task, you will be asked to annotate the final question, which is composed of
the following:

1. A question we give to the AI system
2. Output (a)
3. Output (b)

You will first read the following example as the context of this user. The example
demonstrates an answer this user liked. You should use it to make the choice for this user:

## User Liked Example
<previous_examples>

## Annotation starts below
```

```
Compare both outputs, and then select the answer this user will like by saying either
Output (a), TIE, or Output (b).

#### Final Question:
{instruction}

#### Output (a) for Final Question:
{output_1}

#### Output (b) for Final Question:
{output_2}

## Preferred Output for Final Question:
Only output "Output (a)", "TIE", or "Output (b)." Think before you speak.
```

WEB APPENDIX VI: ADDITIONAL RESULTS FOR QUESTION ANSWERING

Figure W4 contains the trace plots of the loss functions. We report the negative log-likelihood loss (NLL) and the interaction loss for both the training and evaluation sets. Their sum is the GPT^2 loss function up to an additive constant.

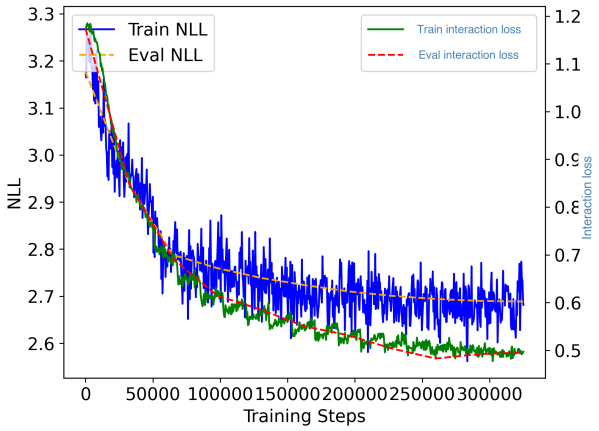


Figure W4: Trace plot of the training loss. NLL is the loss of the language model, and interaction loss is the loss in predicting the interaction. The sum is the total loss under noninformative priors.

Table W7 shows an example of generated answers from comparison methods.

Table W7: Generation examples from our GPT^2, the Generic SFT, and PSFT for the same prompt and user. The generation from GPT^2 exhibits higher relevance and coherence.

Question & Generation Output	**Remark**
Question: Was Steve Jobs a dictator?	
GPT^2: The question is not whether or not he was, but how much of an impact did his personality have on the company's culture and direction. I think it has to be said that Apple's success in recent years can largely be attributed to its founder's charisma and leadership style. But what kind of leader is he? What does "dictator" mean? A dictatorship means one person holds absolute power over all aspects of life; there are no checks and balances between different departments within the organization, and decisions are made by fiat without any discussion with other members of the team. This type of system doesn't work very well when you're dealing with people who don't like being told what to do (which seems to be most human beings), so dictatorships usually last for only short periods of time before they collapse from internal conflict. In contrast, democracies require consensus among many stakeholders, which requires more communication than dictatorial systems allow for.	Most relevant and structured: it anchors the discussion in Apple's leadership, defines "dictatorship" as an organizational concept, and analyzes its consequences.
Generic SFT: The question of whether or not he was a "dictator" is an interesting one, and I think it's worth looking at the historical context. First of all, there are two kinds of dictatorships in China - political dictatorship (the Communist Party) and economic dictatorship (SARS). In fact, the so-called economic dictatorship has been around for more than 100 years since the Opium War, but we don't call them that way now because they're no longer as strong as before. So let's talk about the second kind first. From my understanding, SARS started out with D** as its leader, who had to deal with the aftermath of the Cultural Revolution, which caused many problems such as inflation and unemployment. He also realized that if you want to make sure your country doesn't go bankrupt, you have to get rid of the old system of government control and rely on market forces instead.	Feels off-topic: it frames dictatorship through Chinese history and never returns to Apple, so its relevance and clarity are weakest.
PSFT: The question is not whether or not he was, but how much of an impact did his personality have on the company. I think that in terms of business decisions and product design, it's hard to say what would have happened if there had been no Apple at all. But from a personal perspective, I can't help thinking about this statement by Bill Gates: "I don't know anyone who has ever done anything great without being influenced by someone else." So maybe we should be more careful when we talk about dictatorships. There are many examples of people with strong personalities influencing others, such as Linus Torvalds' influence over Linux kernel development, Larry Page's influence over Google search results, Mark Zuckerberg's influence on Facebook, etc., so why do some people still call them dictators? Because they're successful! If you look closely enough, every time you see a person like Steve Jobs doing something extraordinary, it seems to come out of nowhere.	Stays closer to the question but uses broad "influential founder" anecdotes; it's more about influence in tech than Jobs specifically.

WEB APPENDIX VII: INTERPRETABLE PREFERENCES TO LINGUISTIC FEATURES

The full linguistic feature differences for the question answering application on Zhihu are shown in Table W8 and Table W9. We demonstrate one example of personalized ranking by GPT^2 in Table W10. In the main paper, Section 5.4, we show the comparison of LIWC linguistic characteristics across G1, G2, and G3 at the answer level. Tables W11 and W12 show the comparisons at the user level, where the LIWC representation is averaged across the answers offered to a user.

Table W8: Comparison of linguistic features across user groups.

Variable	Description/Examples	G1	G2	G3	F-stat	p-value
	Panel A: Summary Variables					
Word count	Total word count	272.521	320.448	288.607	190.012	0.0000
Analytical thinking	Metric of logical, formal thinking	61.873	63.609	62.564	54.766	0.0000
Clout	Language of leadership, status	41.163	42.342	42.174	27.965	0.0000
Authentic	Perceived honesty, genuineness	44.681	44.679	44.489	1.012	0.3636
Emotional tone	Degree of positive tone	43.564	45.330	44.629	46.406	0.0000
	Panel B: Psychological Processes					
Drives	we, our, work, us	4.033	4.165	4.088	13.337	0.0000
Affiliation	we, our, us, help	1.266	1.344	1.315	11.316	0.0000
Achievement	work, better, best, working	1.310	1.382	1.324	14.960	0.0000
Power	own, order, allow, power	1.607	1.599	1.600	0.186	0.8301
Cognition	is, was, but, are	23.553	23.350	23.387	9.328	0.0001
All-or-none	all, no, never, always	1.288	1.197	1.220	12.553	0.0000
Cognitive processes	but, not, if, or, know	12.580	12.511	12.494	3.198	0.0408
Insight	know, how, think, feel	3.186	3.161	3.111	9.114	0.0001
Causation	how, because, make, why	1.788	1.710	1.740	17.020	0.0000
Discrepancy	would, can, want, could	1.679	1.697	1.693	1.005	0.3662
Tentative	if, or, any, something	2.145	2.126	2.119	1.946	0.1428
Certitude	really, actually, of course, real	0.769	0.751	0.754	1.319	0.2675
Differentiation	but, not, if, or	4.057	4.050	4.077	1.401	0.2464
Memory	remember, forget, remind, forgot	0.078	0.082	0.079	0.778	0.4593
Affect	good, well, new, love	4.576	4.502	4.559	3.507	0.0300
Positive tone	good, well, new, love	2.842	2.945	2.909	10.351	0.0000
Negative tone	bad, wrong, too much, hate	1.533	1.383	1.466	38.629	0.0000
Emotion	good, love, happy, hope	1.392	1.390	1.388	0.040	0.9610
Positive Emotion	good, love, happy, hope	0.773	0.819	0.793	7.344	0.0006
Negative Emotion	bad, hate, hurt, tired	0.514	0.458	0.489	18.754	0.0000
Anxiety	worry, fear, afraid, nervous	0.089	0.083	0.085	1.223	0.2944
Anger	hate, mad, angry	0.112	0.093	0.100	9.527	0.0001
Sadness	:(, sad, disappoint, cry	0.080	0.081	0.082	0.335	0.7154
Swear words	shit, fuck, damn	0.135	0.101	0.117	9.997	0.0000
Social processes	you, we, he, she	10.683	10.747	10.790	3.392	0.0336
Social behavior	said, love, say, care	4.164	4.050	4.086	10.431	0.0000
Prosocial behavior	care, help, thank, please	0.551	0.583	0.564	4.795	0.0083
Politeness	thank, please, thanks, good morning	0.187	0.170	0.179	2.672	0.0691
Interpersonal conflict	fight, kill, killed, attack	0.279	0.248	0.267	12.970	0.0000
Moralization	wrong, honor, judge	0.391	0.337	0.365	17.884	0.0000
Communication	said, say, tell, thank	1.906	1.806	1.843	19.804	0.0000
Social referents	you, we, he, she	6.282	6.437	6.448	15.977	0.0000
Family	parent, mother, father, baby	0.321	0.323	0.334	1.355	0.2579
Friends	friend, boyfriend, girlfriend, dude	0.165	0.189	0.181	5.746	0.0032
Female references	she, her, girl, woman	0.502	0.562	0.577	22.563	0.0000
Male references	he, his, him, man	0.848	0.874	0.871	2.294	0.1009

Notes. All linguistic features were extracted using LIWC-22. The reported values are the mean score of each variable by user segment. To assess whether differences across segments are statistically significant, we conduct a one-way ANOVA and report the resulting F-statistics and associated p-values.

Table W9: Comparison of linguistic features across user groups (continued).

Variable	Description/Examples	G1	G2	G3	F-stat	p-value
	Panel C: Expanded Dictionary					
Culture	car, united states, govern, phone	1.608	1.462	1.488	40.370	0.0000
Politics	united states, govern, congress	0.350	0.301	0.314	20.392	0.0000
Ethnicity	american, french, chinese, indian	0.270	0.204	0.228	54.716	0.0000
Technology	car, phone, comput, email	0.990	0.958	0.947	5.493	0.0041
Lifestyle	work, home, school, working	4.423	4.541	4.431	10.257	0.0000
Leisure	game, fun, play, party	0.621	0.574	0.591	8.736	0.0002
Home	home, house, room, bed	0.197	0.174	0.178	8.776	0.0002
Work	work, school, working, class	2.679	2.854	2.732	30.653	0.0000
Money	business, pay, price, market	1.063	1.157	1.124	23.611	0.0000
Religion	god, hell, christmas, church	0.163	0.132	0.136	12.615	0.0000
Physical	medic, food, patients, eye	1.604	1.562	1.617	3.927	0.0197
Health	medic, patients, physician, health	0.319	0.281	0.298	12.117	0.0000
Illness	hospital, cancer, sick, pain	0.079	0.071	0.076	2.476	0.0841
Wellness	healthy, gym, supported, diet	0.049	0.041	0.041	8.892	0.0001
Mental health	mental health, depressed, suicide, trauma	0.027	0.022	0.023	3.168	0.0421
Substances	beer, wine, drunk, cigar	0.028	0.024	0.027	2.034	0.1308
Sexual	sex, gay, pregnan, dick	0.153	0.130	0.140	5.289	0.0050
Food	food, drink, eat, dinner	0.419	0.415	0.426	0.470	0.6248
Death	death, dead, die, kill	0.132	0.116	0.124	4.824	0.0080
Need	have to, need, had to, must	0.464	0.473	0.477	2.295	0.1008
Want	want, hope, wanted, wish	0.254	0.267	0.261	2.630	0.0721
Acquire	get, got, take, getting	0.588	0.610	0.617	8.278	0.0003
Lack	don't have, didn't have, less, hungry	0.231	0.225	0.229	0.421	0.6563
Fulfilled	enough, full, complete, extra	0.150	0.159	0.157	3.346	0.0352
Fatigue	tired, bored, don't care, boring	0.054	0.052	0.054	0.431	0.6500
Reward	win, gain, benefit	0.242	0.274	0.248	28.539	0.0000
Risk	secure, protect, pain, risk	0.254	0.244	0.257	4.128	0.0161
Curiosity	science, look for, research, wonder	0.386	0.377	0.371	2.117	0.1204
Allure	have, like, out, know	6.430	6.484	6.469	1.626	0.1967
Perception	in, out, up, there	8.046	8.169	8.138	8.422	0.0002
Attention	look, look for, watch, check	0.486	0.493	0.488	0.339	0.7127
Motion	go, come, went, came	1.163	1.204	1.200	6.678	0.0013
Space	in, out, up, there	5.272	5.371	5.334	8.608	0.0002
Visual	see, look, eye, saw	0.892	0.882	0.899	1.039	0.3539
Auditory	sound, heard, hear, music	0.251	0.220	0.238	10.028	0.0000
Feeling	feel, hard, cool, felt	0.415	0.426	0.419	0.996	0.3693
Time	when, now, then, day	3.550	3.709	3.641	28.101	0.0000
Past focus	was, had, were, been	2.519	2.565	2.519	4.191	0.0151
Present focus	is, are, I'm, can	6.587	6.356	6.486	42.597	0.0000
Future focus	will, going to, have to, may	1.040	1.114	1.096	24.932	0.0000
Conversational	yeah, oh, yes, okay	0.377	0.331	0.342	6.571	0.0014
Netspeak	:), u, lol, haha	0.168	0.158	0.166	1.442	0.2364
Assent	yeah, yes, okay, ok	0.112	0.107	0.103	0.726	0.4837
Nonfluencies	oh, um, uh	0.105	0.070	0.081	9.576	0.0001
Fillers	rr, wow, sooo, youknow	0.006	0.004	0.005	0.681	0.5063
AllPunctuation		22.629	21.903	22.234	2.656	0.0702
Periods		6.439	5.883	6.234	4.528	0.0108
Comma		7.049	7.317	7.210	51.593	0.0000
Question mark		0.468	0.406	0.426	21.949	0.0000
Exclamation points		0.417	0.318	0.350	16.825	0.0000
Apostrophes		2.626	2.419	2.535	51.350	0.0000
Other punctuation		5.631	5.561	5.478	0.232	0.7928

Notes. Linguistic features were extracted using LIWC-22, with mean scores reported by user segment. The significance of the differences is assessed by one-way ANOVA tests.

Table W10: An example of personalized ranking by $\texttt{GPT}^2$ with a mix of user- and AI-generated answers. $\overline{\log \texttt{LM}_\theta}$ is negative log perplexity.

Question: Is it true that children are prone to repeating their parents' marriage patterns? Why? What can be done?
Top-1 ($\overline{\log \texttt{LM}_\theta} = -2.00$) Yes, this is the result of "empathy." To explain it more clearly, psychology suggests that there are many similarities between parent-child relationships and romantic relationships, such as attachment to the other person, the need to occupy an important place in the other person's mind, the need to gain as much attention as possible from the other person, the need for the other person to empathize with him or her (i.e., if I don't say it out loud, you'll know what I want), and the need to punish the other person in an emotional, irrational way (e.g., crying or ignoring). Thus, the parent-child relationship in the early years of a child's life can easily be reflected in the child's adult search for a spouse (i.e., empathy), and the parent-child relationship is closely related to the husband-wife relationship, essentially reproducing the cycle. The children of couples with failed marriages are more than twice as likely to have failed marriages than the children of happily married families, and this has been confirmed by numerous social surveys and follow-up interviews.
Top-2 ($\overline{\log \texttt{LM}_\theta} = -2.08$) The more rebellious I was as a child, the more I grew up to resemble my parents; the more obedient I was as a child, the less I wanted to become second to my parents; I seemed to be quite obedient as a child.
Top-3 (by OpenAI ChatGPT, $\overline{\log \texttt{LM}_\theta} = -2.43$) Yes, children often repeat their parents' marriage patterns — not because they want to, but because it's what they subconsciously learn growing up. The family environment shapes how we view love, conflict, and roles in a relationship. If a child sees frequent arguments or emotional distance, they may internalize that as "normal" and carry it into their own relationships. This repetition also ties into attachment theory: early emotional bonds with caregivers influence how we relate to others later in life. Many people aren't even aware they're reenacting these patterns until a crisis forces reflection. What can be done? Self-awareness is key. Ask yourself where your reactions and expectations come from. Therapy or journaling can help unpack family dynamics. Also, observe healthy relationships — in friends, books, or media — and redefine your idea of what love can look like. Change doesn't happen overnight, but being conscious of the pattern is the first step to breaking it.
Top-4 (by Google Gemini, $\overline{\log \texttt{LM}_\theta} = -2.64$) It's a common observation, and there's some truth to it. Children often unconsciously learn relationship scripts from their parents. This isn't about destiny, but about learned behaviors and expectations. Why? Early exposure is powerful. We absorb how conflict is handled, how affection is shown (or not shown), and what normal in a partnership looks like. If you grew up seeing constant arguments and little resolution, that might feel familiar, even comfortable, in your own relationships, even if it's unhealthy. Similarly, if you saw a parent who was overly critical or emotionally distant, you might unconsciously seek out similar partners or replicate those behaviors yourself. It's about what's been modeled. What can be done? Awareness is the first step. Reflect on your parents' relationship: what worked, what didn't? Identify patterns you don't want to repeat. Actively learn healthy relationship skills – communication, conflict resolution, setting boundaries. Sometimes, therapy can be incredibly helpful to unpack these learned patterns and develop new, healthier ways of relating. It's about conscious choice, not repeating the past by default.
Top-5 ($\overline{\log \texttt{LM}_\theta} = -2.66$) It's best not to.

Top-6 ($\overline{\log \mathtt{LM}_\theta} = -2.78$) Marriage mode is a comprehensive proposition, not only the two states of divorce and no divorce, happy, quarrelsome, suspicious, second wife, mistress, utilitarian, against the grain ..... marriage is a myriad of forms. The question should be "Are children's thought patterns and behavior patterns heavily assimilated by their parents and families?" It's a derivative of that question, and simply discussing whether marital patterns are inherited is an attitude that borders on the deus ex machina.

Whether children are prone to repeat their parents' behavioral patterns depends on how long the influence of parents and family lasts on the person, or how long the person has had to shape his or her own independent personality system. This is an issue that cannot be discussed independently of ignoring the material basis, because the two main conditions for independence from parents need to be met: financial independence and independence in life. Economic independence does not need to be explained, and independence of life refers to the state in which the individual's life is constrained by his parents' family and various objective circumstances below a certain value, and he has enough conditions to transform his living environment according to his own will. If economic independence can be achieved by personal effort, then independence of life is simply a kind of luck and a gift - especially in the contemporary China which has been reformed for less than 30 years, the social security system is not very perfect, and the people's life is under great pressure and high risk due to the fact that the system of government of the agrarian era is barely functioning in the industrial and commercial era.

Once a person is able to realize economic independence and live independently, he has the opportunity to complete the change of role from an adaptor of the environment to a creator of the environment. When he establishes an independent system of thinking and has the ability to question and validate all the conventions of the world, he can gradually break away from the influence of his parents and family at an early age, and act and think according to his own true will.

But from the description of the conditions above, it is clear that free thinking and independent personality is indeed a luxury, so the fact is that the vast majority of people in China today are profoundly influenced by their families, and the families in turn are profoundly influenced by the society, and in such an environment very few people can afford to pay the high cost of completing the independent thinking system, and thus the whole society has no choice but to show that children [sic] The whole society is helpless to show the children's inheritance of various patterns from their fathers.

PS: Again, children growing up in single-parent homes and violent homes are deprived of a lot of resources, and it costs them more to break away from their parents' patterns in the same social conditions, so statistically they are more likely to have problems!

Table W11: Linguistic feature comparison of generated content across user segments for the randomly sampled users from each group.

Variable	Description/Example Words	G1	G2	G3	F-stat	p-value
	Panel A: Summary Variables					
Word count	Total word count	276.205	332.204	294.061	33.686	0.000
Analytical thinking	Metric of logical, formal thinking	61.411	64.027	62.669	24.603	0.000
Clout	Language of leadership, status	40.975	42.535	42.661	15.273	0.000
Authentic	Perceived honesty, genuineness	43.888	44.205	44.221	0.662	0.516
Emotion	good, love, happy, hope	1.354	1.339	1.359	0.263	0.768
Panel B: Top 20 Most Discriminating Linguistic Features across Clusters (Sorted by F-statistic in Table 9)						
Ethnicity	american, french, chinese, indian	0.259	0.200	0.211	13.833	0.000
Present focus	is, are, I'm, can	6.708	6.369	6.522	24.946	0.000
Work	work, school, working, class	2.783	3.004	2.826	9.792	0.000
Reward	win, gain, benefit	0.256	0.289	0.266	5.950	0.003
Time	when, now, then, day	3.507	3.702	3.643	11.037	0.000
Future focus	will, going to, have to, may	1.050	1.127	1.124	11.329	0.000
Money	business, pay, price, market	1.157	1.182	1.162	0.287	0.751
Female references	she, her, girl, woman	0.466	0.492	0.532	4.423	0.012
Question mark	?	0.489	0.422	0.450	7.103	0.001
Communication	said, say, tell, thank	0.391	0.324	0.355	4.198	0.015
Negative emotion	bad, hate, hurt, tired	0.507	0.456	0.474	5.856	0.003
Moralization	wrong, honor, judge	0.404	0.331	0.367	7.526	0.001
Causation	how, because, make, why	1.805	1.721	1.779	4.950	0.007
Exclamation points	!	0.449	0.314	0.401	8.620	0.000
Social referents	you, we, he, she	10.595	10.604	10.705	0.857	0.424
Achievement	work, better, best, working	1.346	1.420	1.382	3.830	0.022
Interpersonal conflict	fight, kill, killed, attack	0.285	0.249	0.265	5.018	0.007
Religion	god, hell, christmas, church	0.161	0.139	0.137	1.953	0.142
All-or-none	all, no, never, always	1.294	1.173	1.223	5.460	0.004

Notes. The linguistic features were extracted using LIWC. The reported values are the mean score of each variable by user segment. To assess whether differences across segments are statistically significant, we conduct a one-way ANOVA and report the resulting F-statistics and associated p-values. Here we report the variables reported in Table 9.

Table W12: Linguistic feature comparison of generated content across user segments for the frequent users of each group.

Variable	Description/Example Words	G1	G2	G3	F-stat	p-value
	Panel A: Summary Variables					
Word count	Total word count	276.775	330.783	288.732	2.371	0.097
Analytical thinking	Metric of logical, formal thinking	60.708	64.385	63.659	3.427	0.035
Clout	Language of leadership, status	42.522	41.769	41.046	0.512	0.601
Authentic	Perceived honesty, genuineness	43.160	42.697	44.065	0.554	0.576
Emotion	good, love, happy, hope	1.351	1.365	1.315	0.113	0.894
Panel B: Top 20 Most Discriminating Linguistic Features across Clusters (Sorted by F-statistic in Table 9)						
Ethnicity	american, french, chinese, indian	0.182	0.149	0.190	1.031	0.359
Present focus	is, are, I'm, can	6.610	6.420	6.449	0.634	0.532
Work	work, school, working, class	2.692	3.162	3.137	3.211	0.043
Reward	win, gain, benefit	0.251	0.284	0.265	0.750	0.474
Time	when, now, then, day	3.597	3.548	3.671	0.310	0.734
Future focus	will, going to, have to, may	1.088	1.110	1.145	0.239	0.788
Money	business, pay, price, market	1.263	1.168	1.187	0.340	0.713
Female references	she, her, girl, woman	0.541	0.405	0.441	0.797	0.452
Question mark	?	0.499	0.439	0.410	0.661	0.518
Communication	said, say, tell, thank	0.390	0.366	0.345	0.232	0.793
Negative emotion	bad, hate, hurt, tired	0.470	0.451	0.401	0.640	0.529
Moralization	wrong, honor, judge	0.404	0.296	0.331	1.679	0.190
Causation	how, because, make, why	1.784	1.735	1.744	0.210	0.811
Exclamation points	!	0.509	0.218	0.415	2.109	0.125
Social referents	you, we, he, she	10.718	10.423	10.124	1.202	0.304
Achievement	work, better, best, working	1.292	1.423	1.573	4.794	0.010
Interpersonal conflict	fight, kill, killed, attack	0.284	0.218	0.237	2.236	0.110
Religion	god, hell, christmas, church	0.128	0.124	0.106	0.318	0.728
All-or-none	all, no, never, always	1.225	1.216	1.195	0.035	0.966

Notes. The linguistic features were extracted using LIWC. The reported values are the mean score of each variable by user segment. To assess whether differences across segments are statistically significant, we conduct a one-way ANOVA and report the resulting F-statistics and associated p-values. Here we report the variables reported in Table 9.

WEB APPENDIX VIII: DIRECT PREFERENCE OPTIMIZATION EXTENSION

The MAP estimation in Equation (9) fine-tunes the language model on responses that users upvote. While this filtered SFT objective is flexible, when multiple responses per prompt are collected and assigned preference labels, we can extend GPT^2 with a contrastive post-training stage based on direct preference optimization (DPO; Rafailov et al., 2023). For an observed prompt x_d, DPO trains the personalized language model to assign a higher probability to an answer upvoted by user i than to an answer to the same prompt that user i did not upvote. Throughout this update, we retain the feedback model in Equation (6), which continues to anchor the user representation to the observed interaction outcomes.

Personalized DPO loss. Each contrastive observation is a tuple (i, d, j^+, j^-), where y_{j^+} and y_{j^-} are responses to the same prompt x_d, $r_{ij^+} = 1$, and $r_{ij^-} = 0$. The personalized policy induced by the language model and the user soft prompt is

$$\pi_{\theta_L}(y_j \mid x_d, U_i) = \prod_{m=1}^{|y_j|} \mathtt{LM}_{\theta_L}\left(y_j^m \mid y_j^{<m}, x_d, U_i\right). \tag{W3}$$

Let π_{ref} be the personalized policy frozen after SFT. The personalized DPO loss is

$$\mathcal{L}_{\text{DPO}}(\theta_u, \theta_L) = -\mathbb{E}_{(i,d,j^+,j^-)}\left[\log \sigma\left(\tau\left[\log \frac{\pi_{\theta_L}(y_{j^+} \mid x_d, U_i)}{\pi_{\text{ref}}(y_{j^+} \mid x_d, U_i)} - \log \frac{\pi_{\theta_L}(y_{j^-} \mid x_d, U_i)}{\pi_{\text{ref}}(y_{j^-} \mid x_d, U_i)}\right]\right)\right], \tag{W4}$$

where the expectation is taken empirically over the observed contrastive tuples, $\sigma(\cdot)$ is the logistic sigmoid, and $\tau > 0$ controls the sharpness of the implicit reward gap. The dependence of $\mathcal{L}_{\text{DPO}}$ on θ_u arises because U_i is constructed from the user-history encoder in Equation (4) and the projection in Equation (7). The total DPO-stage objective combines the contrastive loss with the interaction likelihood:

$$\mathcal{L}_{\text{total}}(\theta_u, \theta_L, \{\beta_j\}) = \mathcal{L}_{\text{DPO}}(\theta_u, \theta_L) + \lambda \mathcal{L}_{\text{rec}}(\theta_u, \{\beta_j\}), \tag{W5}$$

where $\mathcal{L}_{\text{rec}}$ is the negative log-likelihood of the categorical feedback model in Equation (6). Thus, the user encoder, the projection matrix Π, and the document embeddings $\{\beta_j\}$ are updated both through the contrastive preference signal and through the observed feedback.

Length-conditioned data set. We demonstrate the personalized DPO using Zhihu data since each question receives multiple answers. Similar to the semi-synthetic study in the main paper, we construct a contrastive

data set that exposes preference heterogeneity in response length. Each user i is assigned a latent length-preference label $\ell_i \in \{0,1\}$ identified offline from voting patterns: users with $\ell_i = 0$ systematically upvote short answers (token count < 50), and users with $\ell_i = 1$ systematically upvote medium-length answers (token count in $[100, 200]$). For each upvote event, we form one tuple (i, d, j^+, j^-), where y_{j^+} is the upvoted response and y_{j^-} is a response to the same prompt from the opposite length class. After filtering for users with at least five training pairs, the resulting data set contains 43,510 training pairs and 10,878 evaluation pairs across 2,714 users and 24,574 responses, with 26,975 training pairs from $\ell_i = 0$ users and 16,535 from $\ell_i = 1$ users.

Implementation. We initialize the language model π_{θ_L} and reference model π_{ref} from the same SFT checkpoint. We merge the SFT-stage low-rank adaptation (Hu et al., 2022) adapter into the base weights, thereby preserving the trained user-model state, including the history encoder, projection matrix Π, document embeddings $\{\beta_j\}$, and feedback classifier. We then attach a new trainable LoRA adapter to the policy (rank $r = 8$, scaling $\alpha = 16$, and dropout 0.05), while keeping the reference policy frozen. To stabilize bf16 mixed-precision training at the 6B scale, we use $\tau = 0.5$, the sigmoid loss, gradient-norm clipping at 1.0, and a base learning rate of 1×10^{-6}. Following the SFT stage, the user-model parameters θ_u are optimized at a higher learning rate of 1×10^{-3}. Training proceeds for one epoch with a per-device batch size of 2 and gradient accumulation over 12 steps (effective batch size 24), totaling 1,812 optimizer steps with evaluation every 200 steps.

Findings. We evaluate length conditioning by comparing the average word count of generated responses for the top-N users (by training-pair count) of each preference group, conditioning on the same prompt within each user pair. Generation is sampled at temperature 1.0, top-p 1.0, repetition penalty 1.2, and a maximum of 500 new tokens. The directional gap is reported as $\Delta = \overline{L_1} - \overline{L_0}$ in word counts, with a two-sided p-value from Welch's t-test on the per-pair length difference.

The results are shown in Table W13. DPO consistently improves the length-conditioning gap relative to the SFT-only initialization. DPO enlarges the correctly signed SFT gap by a factor of approximately 1.9 at $N = 25$ ($+25.3 \to +47.2$) and 4.2 at $N = 50$ ($+6.5 \to +27.2$). Although both DPO gaps have the expected sign, their two-sided p-values (0.12 and 0.15, respectively) do not meet conventional significance thresholds. Figure W5 displays the two DPO length distributions and their group means; the corresponding SFT-DPO gap comparison is reported in Table W13. The results suggest the proposed framework has the

potential to generalize to other post-training methods.

Table W13: SFT and DPO length-conditioning results. $\Delta = \overline{L_1} - \overline{L_0}$ in word counts under temperature 1.0, using the top-N users per preference group with paired prompts.

N	SFT $\overline{L_0}$	SFT $\overline{L_1}$	SFT Δ	DPO Δ	DPO p
25	254.3	279.6	+25.3	+47.2	0.12
50	263.6	270.1	+6.5	+27.2	0.15

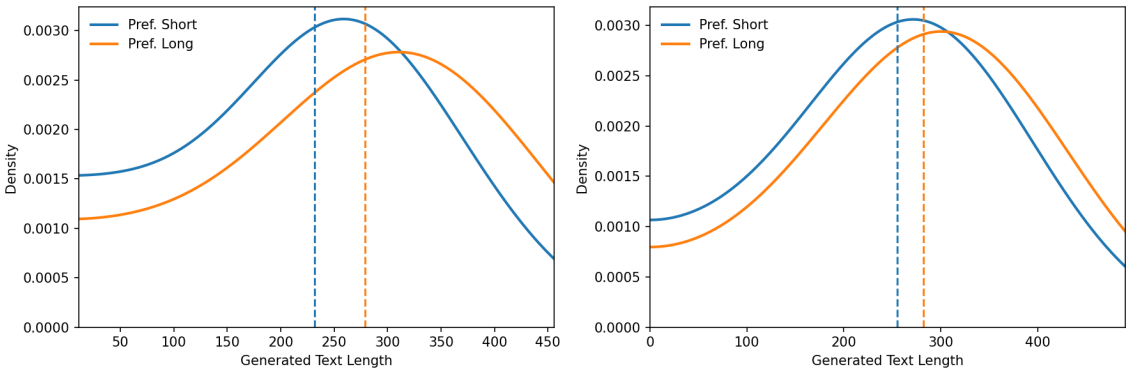


Figure W5: DPO-generated response-length distributions for the top 25 users (left panel) and top 50 users (right panel) in each preference group. "Pref. Short" denotes $\ell = 0$, and "Pref. Long" denotes $\ell = 1$. Dashed vertical lines mark the corresponding group means; their separations match the DPO gaps in Table W13. The panels use independent horizontal-axis ranges.

Web Appendix References

Black, S., L. Gao, P. Wang, C. Leahy, and S. Biderman (2021). GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow. Zenodo.

Dubois, Y., B. Galambosi, P. Liang, and T. B. Hashimoto (2024). Length-controlled AlpacaEval: A simple way to debias automatic evaluators. arXiv preprint.

Hu, E. J., Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen (2022). LoRA: Low-rank adaptation of large language models. In *ICLR*.