Multimodal Technologies and Interaction

MDPI

Article

Choice Vectors: Streamlining Personal AI Alignment Through Binary Selection

Eleanor Watson [1,*], Minh Nguyen [2], Sarah Pan [3] and Shujun Zhang [1]

[1] School of Computing and Engineering, University of Gloucestershire, The Park, Cheltenham GL50 2RH, UK; szhang@glos.ac.uk
[2] Lee Kong Chian School of Business, Singapore Management University, 81 Victoria St, Singapore 188065, Singapore; minh1228@gmail.com
[3] Electrical Engineering & Computer Science Department, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139-4307, USA; sarahpan@mit.edu
* Correspondence: eleanorwatson@connect.glos.ac.uk

Abstract: Value alignment for AI is not "one-size-fits-all": even polite and friendly models can still fail to represent individual user contexts and preferences, and local cultural norms. This paper presents a modular workflow for personal fine-tuning, synthesizing four core components from our previous research: (1) robust vectorization of user values and preferences, (2) a binary choice user interface (UI) approach to capturing those preferences with minimal cognitive load, (3) contrastive activation methods for steering large language models (LLMs) via difference vectors, and (4) knowledge graph integration for more auditable and structured alignment. Our approach—descended from past research on "Towards an End-to-End Personal Fine-Tuning Framework"—demonstrates how these elements can be combined to create personalized, context-rich alignment solutions. We report on user studies for the forced-choice UI, describe an experimental pipeline for deriving "control vectors", and propose a "moral graph" method for bridging symbolic and vector-based alignment. Our findings suggest that multi-pronged personalization can significantly reduce user annotation fatigue, improve alignment fidelity, and allow for more flexible, interpretable AI behaviors.

Keywords: AI alignment; value vectorization; binary choice UI; contrastive activation; knowledge graph; personal fine-tuning

Academic Editors: Wei Liu, Jan Auernhammer and Takumi Ohashi

Received: 15 January 2025
Revised: 25 February 2025
Accepted: 27 February 2025
Published: 3 March 2025

Citation: Watson, E.; Nguyen, M.; Pan, S.; Zhang, S. Choice Vectors: Streamlining Personal AI Alignment Through Binary Selection. *Multimodal Technol. Interact.* **2025**, *9*, 22. https://doi.org/10.3390/mti9030022

1. Introduction

Large language models (LLMs) are increasingly deployed in contexts where they must cater to each user's unique preferences. Ensuring reliable alignment with those individual preferences—what we term personal alignment—remains a significant challenge. Problems arise when global, one-size-fits-all safety or style guidelines override a user's specific needs. Such misalignment can pose agentic safety risks, such as neglecting a user's dietary restrictions or failing to consider accessibility requirements.

Building on previous work exploring user preference collection [1], this paper presents a flexible pipeline for personal alignment that weaves together four complementary components. Although Reinforcement Learning from Human Feedback (RLHF) has improved the behavior of LLMs in a broad sense, it does not always capture the nuanced or evolving priorities of individual users. This gap motivates our three main research questions:

RQ1: How can we gather each user's preferences without creating undue cognitive overhead?

RQ2: Which technical mechanisms ensure a reliable translation of user preferences into model behavior?

RQ3: How can we maintain transparency and interpretability when tailoring AI systems to individuals?

We make the following specific contributions:

1. A binary choice interface that reduces cognitive load yet elicits rich personal preference data.
2. A hybrid vector–symbolic architecture that bridges continuous embeddings with more interpretable symbolic representations.
3. An extensible pipeline that supports personal alignment through modular and complementary methods.
4. Empirical validation via a small pilot study.

Our approach synthesizes four key components:

- Value Vectorization and Representation Engineering: We show how to extract user values as low-dimensional embeddings or "control vectors", including forward-pass-based methods and contrastive approaches.
- Binary Choice UI: We introduce a minimal-friction forced-choice interface, inspired by casual quiz mechanics, that captures user stances and reduces survey fatigue.
- Contrastive Activation Methods: We detail how difference vectors can be derived from pairs of user-labeled extremes (e.g., "formal vs. casual", "left vs. right") and used to modulate outputs.
- Knowledge Graph Integration: We describe a "moral graph" approach to store user preferences in a structured, auditable form, bridging vector-based alignment with symbolic inference.

While each component has precedent in existing work, their combination into a coherent pipeline for personal alignment represents a novel contribution to the field.

2. Related Work

Research on alignment for large language models has often prioritized global frameworks, most notably Reinforcement Learning from Human Feedback (RLHF). RLHF has enabled models such as ChatGPT to become polite or helpful, but has also led to critiques that it imposes homogenized norms that might not reflect certain cultural or personal contexts [2,3]. More recent extensions, such as constitutional AI, similarly focus on deriving sets of universal or near-universal principles.

In parallel, a smaller but growing body of literature emphasizes personalized or "local" alignment. Direct Preference Optimization does this in an end-to-end fashion, implicitly learning the preferable traits from the unwanted ones through carefully curated preference datasets.

There has also been increased attention to "representation engineering", which can introduce or amplify specific traits within the model's latent representations within a model. For example, the concept of "activation addition" harnesses the difference between hidden states that correspond to two extremes. Methods for controlling generation by injecting difference vectors at specific layers are gaining traction, especially given the constraints that end-to-end fine-tuning might impose.

User interfaces that engage in preference elicitation range from full chat logs—wherein a user effectively "teaches" the model through repeated conversation—to more structured questionnaire-based approaches. Our earlier prototypes, referred to as Mark 1 and Mark 2, adopted a chat or extended survey approach, but we observed significant user drop-off

due to the cognitive overhead required. This led us to investigate simpler forced-choice designs reminiscent of personality quizzes, which appear to foster less fatigue.

Symbolic approaches to AI alignment, such as knowledge graphs or moral graphs, attempt to capture user value systems in the form of discrete nodes and edges that define the relationships among values [4]. This has the advantage of interpretability and can allow the system to reason explicitly about conflicts between user preferences. However, purely symbolic approaches can be brittle, particularly for language-generation tasks that rely on high-level distributional representations of style and context. For that reason, bridging symbolic moral graphs with continuous vector spaces is increasingly seen as an intriguing direction, potentially offering the "best of both worlds" in explainability and generative flexibility.

Comparison with RLHF and DPO

Although Reinforcement Learning from Human Feedback (RLHF) and Direct Preference Optimization (DPO) have significantly advanced alignment for large language models, they primarily focus on shaping models at a global or population-wide level [5,6]. In contrast, Choice Vectors address the need for hyper-personalization by providing lightweight, user-specific adjustments that do not require extensive retraining or large preference datasets. To highlight key distinctions, Table 1 compares the Choice Vectors pipeline with RLHF and DPO along several dimensions, including training overhead, user burden, interpretability, and real-time adaptability.

Table 1. Comparison of RLHF, DPO, and Choice Vectors.

Aspect	RLHF	DPO	Choice Vectors
Training Overhead	High; requires curated human feedback data and iterative tuning	Moderate; depends on preference datasets but still requires custom training	Minimal if using difference vectors for real-time steering
User Involvement	Often indirect, via external annotators	Can use user-labeled data, but not typically on-the-fly	Direct forced-choice or sample-labeled data; immediate feedback
Interpretability	Limited unless reward model is interpretable	Moderate; can track preference patterns	High if combined with knowledge graphs ("moral graph")
Real-Time Adaptation	Low; RLHF-based behavior is fairly fixed post-training	Low–moderate, re-optimization required	High; can inject difference vectors at inference time
Use Case Focus	Broad safety, general helpfulness	Preference-driven large dataset tasks	Personalized style, tone, moral or ethical stances

The Choice Vectors pipeline thus provides an alternative route to local alignment, maintaining a complementary relationship with RLHF- or DPO-based global guardrails.

3. Proposed Approach

The approach we propose weaves together four elements into a cohesive pipeline that can be deployed incrementally or as a whole. The result is a way to capture user stances via minimal binary selection, transform these stances into numeric representations

or difference vectors, refine or combine them via contrastive methods, and optionally store them in a knowledge graph for ongoing interpretability and expansion.

3.1. Value Vectorization and Representation Engineering

A key premise is that user values and stylistic preferences can be encoded in vectors that can be inserted into a model's hidden states or prompts. Rather than relying on a monolithic "persona prompt", we aim to isolate particular latent directions that correspond to a user's stances on humor, formality, directness, or any number of ethical and stylistic dimensions.

To achieve this, we collect user examples of "approved" and "disapproved" outputs. For each example, we conduct forward passes in the model, capturing hidden states at certain layers. We then compute either the average difference between states that the user endorses and those the user rejects, or apply a form of dimensionality reduction to discover principal axes of difference.

We employed Principal Component Analysis (PCA) due to its linear interpretability—difference vectors discovered in principal component space can be easily added or subtracted at inference to manipulate model behavior. Alternative methods like t-SNE or UMAP are highly effective for visualizing clusters in high-dimensional data, but they introduce nonlinear transformations that complicate real-time vector addition. These methods excel at revealing complex cluster structures in user embeddings or model activations, helping stakeholders observe how various preferences group or overlap. However, because t-SNE and UMAP optimize for local rather than strictly linear neighborhood preservation, they are less suited for direct manipulation of hidden states.

In other words, we cannot simply "add a UMAP difference vector" at inference time with the same reliability we enjoy using PCA-based offsets. Nevertheless, t-SNE or UMAP may be employed in post hoc analyses to identify emergent clusters of user preferences, confirm that desired styles form coherent regions in the latent space, or explore potential edge cases in preference distributions—thereby complementing the operational benefits of a purely linear approach.

Furthermore, linear methods such as PCA align naturally with "activation addition" [7], preserving meaningful directions in the latent space. Future work could evaluate manifold-based methods for more nuanced preference embeddings, but PCA suffices for our immediate goal: identifying robust, distinct directions for user-specified traits.

The effectiveness of these representation engineering techniques has been dramatically illustrated by Vogel [8], who demonstrated with Mistral-7B how targeted vector manipulations can induce specific cognitive states in language models. By isolating and manipulating activation vectors representing conceptual states (such as 'dream-like' or 'analytical' thinking), Vogel showed how even relatively small modifications to hidden representations can produce significant, predictable shifts in a model's generation style and reasoning approach while maintaining coherent outputs.

Another technique involves prompting the model to produce two extremes of a scenario—for instance, a very compassionate version vs. a very blunt version—and then deriving the difference in hidden-state activations. The result is a "control vector" that can be added to, or subtracted from, the hidden states of new prompts, thereby steering the model's generation style.

We find that such a control vector is lightweight to store and often robust to modest changes in the prompt or the sampling temperature. Though advanced users might want to refine it further through low-rank adapters, for many purposes these difference vectors can be reapplied in real time at inference, without extensive additional training.

3.2. Binary Choice User Interface

While a variety of questionnaires or chat-based interactions can extract user preferences, we introduce a minimal interface in which users are presented with two statements and asked which they prefer or agree with more strongly. This design, which might resemble personality quizzes or short polls, dramatically reduces cognitive load compared to free-form dialogue or multi-question surveys.

Once the user selects one of two statements, the system records a small "offset" in the relevant dimension. For example, a user might be confronted with the choice between "I find sarcastic humor acceptable in almost all contexts" vs. "I prefer empathetic, sincere language even when critical". Choosing the former might nudge the sarcasm dimension upward, while the latter might tilt the system more toward sincerity. Users may be presented with approximately ten such pairs, drawn from a pool that covers style, moral stances, or boundary conditions. The forced-choice model ensures that each decision is quite rapid: in pilot tests, participants reported a high completion rate and a clearer sense of the system's immediate purpose.

These data can then be aggregated in a vector-based manner by associating each forced-choice with a known dimension, or used to define symbolic edges or tags in a knowledge graph. We have found that this binary choice interface yields a set of user stances that can be readily integrated into vector offset approaches, thus forming a streamlined route from minimal user input to robust model transformations.

3.3. Contrastive Activation Methods

Having established how we gather user stances (either from examples or forced choices) and how we can transform them into vectors, we further explore contrastive activation. The general principle is that if one obtains two contrasting states in the model—an extreme scenario and its opposite—then the difference in their hidden states can serve as a "steering vector". For instance, if a user wants the model to sound more "warm and empathetic", we can prompt the model to produce a "warm" response and a "cold" response, then compute the difference at a certain layer. Adding that difference vector to new contexts encourages the warm style to emerge.

Technically, we gather two outputs by prompting the model with instructions that ask it to adopt opposing styles or positions. We record hidden states for each token in each sequence, then average them in a consistent manner. The difference in these averaged states is saved as a small matrix or vector. At inference time, we inject this difference into other prompts by modifying the hidden state at the same layer index. In pilot trials with GPT-2 and a 7B-parameter LLaMA, these methods produce consistent style changes for a large fraction of test prompts, though some user instructions or system-level constraints can override them.

This approach has been further validated by Panickssery et al. [9], who demonstrated a robust implementation of contrastive activation addition specifically for LLaMA 2 models. Their work showed that targeted steering vectors derived from contrastive examples could effectively modify the model's response style while maintaining coherence and factual accuracy across diverse prompts.

To refine our understanding of user preferences, we collect both positive and negative examples of desired model behavior. By analyzing the difference between these examples, we identify the key dimensions along which the model's outputs should vary. This process involves averaging the representations of positive and negative examples separately, then computing their difference to create a steering vector.

To ensure these vectors capture meaningful variation rather than noise, we apply dimensional reduction techniques to identify the primary axes of difference between preferred and non-preferred outputs.

Contrastive activation methods are especially promising because they minimize overhead. They do not require elaborate new training steps. Instead, they rely on the user's willingness to provide or verify a pair of extremes. Since this also integrates well with the forced-choice UI, we can imagine a scenario where the user's preference is gleaned from multiple binary selections, which yields a sense of the "extreme" the user most wants to emulate, thereby reinforcing that difference vector for future output generation.

Another recent approach, Contrastive Preference Learning (CPL) [10], proposes learning optimal policies directly from user preferences, without the intermediary step of reward modeling. CPL leverages a contrastive learning objective, which has shown promise in scaling to large datasets. This approach offers an alternative path to aligning AI systems with human preferences.

Recent work on self-rewarding language models suggests another promising direction for preference refinement [11]. Rather than relying solely on external feedback or static reward models that may be bottlenecked by human performance levels, a self-rewarding approach enables the model to generate its own training signals during preference learning. When applied to our contrastive activation framework, this could allow the system to iteratively refine its understanding of user preferences by generating and evaluating potential responses against learned preference vectors. However, care must be taken to ensure that such self-generated rewards remain aligned with the user's actual preferences rather than drifting toward model-internal optimizations.

3.4. Knowledge Graph/Moral Graph Integration

Eliciting and representing human values for AI alignment poses significant challenges. Values are often abstract, context-dependent, and difficult to articulate. Recent work has proposed representing values as "attentional policies" that capture what people pay attention to when making meaningful choices. This approach aims to make values more concrete, disambiguated, and ultimately more useful for aligning AI systems.

Although vector-based methods are powerful and relatively easy to implement, they remain somewhat opaque when it comes to interpretability. Symbolic approaches, such as knowledge graphs, can provide a more explicit representation of the user's moral or stylistic stances.

Knowledge graphs are data structures that store information in a graph format, with entities (nodes) connected by relationships (edges). This structure is highly effective for representing complex networks of information and enabling semantic queries and reasoning over data. In the context of large language models (LLMs), vectors often represent text data features or embeddings in a multidimensional space that captures semantic meaning.

By mapping abstract, high-dimensional data into a knowledge graph, it is possible to enable an LLM to navigate or "traverse" this graph like a tree. This traversal allows the LLM to reason with the data in a structured and potentially more transparent manner, making inferences or drawing conclusions based on the relationships and entities within the graph, rather than solely relying on patterns within the vector space.

It is therefore feasible to derive meaningful, discrete pieces of information from the vectors or embeddings generated by LLMs. This step is crucial for constructing a knowledge graph from vector data, as it identifies which aspects of the vectors should be represented as entities and relationships within the graph. A knowledge graph could therefore allow an LLM to reason with the values by traversing the tree, enhancing the model's ability to interpret and interact with the underlying data.

A knowledge graph can store nodes that reflect particular preferences or constraints—"User generally disapproves of sarcastic jokes", or "User highly values direct honesty"—and edges that capture relationships between these stances, such as contexts in which they apply or their relative priority.

For instance, suppose a user node in the moral graph has edges capturing the stance "prefers honesty over politeness in direct messages." In parallel, we store a corresponding reference vector derived from user-labeled exemplars of "honest but slightly blunt" text. When a new prompt requires the model to navigate between politeness and honesty, the system can consult this node to retrieve both (1) the symbolic declaration that honesty has higher priority for direct messages and (2) the associated vector offset that shifts generation toward directness. This dual approach—symbolic for interpretability, vector-based for real-time style control—helps ensure the model's output remains aligned with the user's explicitly stored preference.

The foundation of our approach is the transformation of user preferences into numerical representations that can influence model behavior. When a user expresses a preference, we capture it as a vector in a high-dimensional space where similar preferences cluster together, as shown in Figure 1. These vectors are normalized to ensure consistent influence across different contexts.

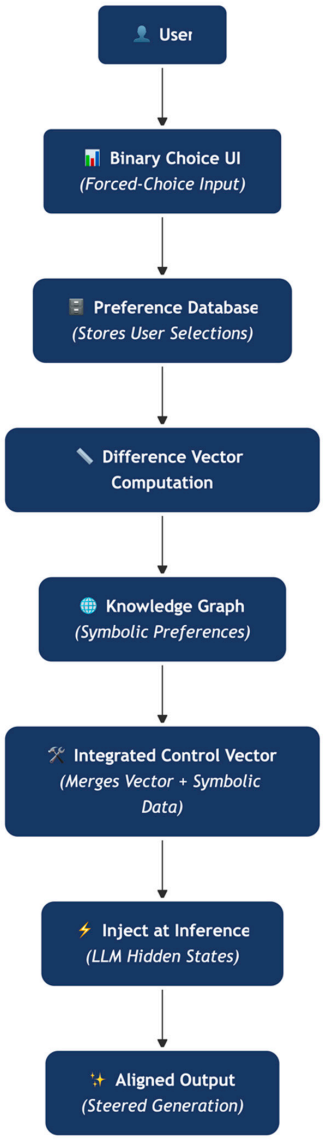

Figure 1. Contrastive activation method for extracting and applying user preference vectors.

The knowledge graph component provides a structured representation of user preferences and their relationships. Each node in the graph represents a specific preference, while edges represent relationships between preferences. These relationships might indicate reinforcement, conflict, or contextual dependencies.

The system maintains a bidirectional mapping between the graph structure and the vector representations. This allows us to combine the interpretability of symbolic representations with the flexibility of continuous vectors. When making decisions about model outputs, the system can consider both the immediate vector representations and the broader context provided by the knowledge graph.

For contrasting preferences—such as formal vs. casual communication styles—we compute a difference vector. This difference represents the direction of change needed to shift model outputs toward one preference or the other. The strength of this shift can be adjusted through a scaling parameter, allowing for fine-tuned control over how strongly the preference affects model behavior.

Combining knowledge graphs with vector-based offsets allows for both interpretability (symbolic) and flexibility (continuous). However, certain trade-offs arise. A symbolic representation (e.g., "User disallows sarcasm in formal writing") is immediately understandable. Purely vector-based adjustments might obscure the rationale for an output style. Knowledge graph constraints must be carefully enumerated. Vectors, meanwhile, can handle subtle emergent behaviors without explicit rule definitions, but risk losing clarity on why the model shifted. Graph structures require updates when preferences evolve, else or conflicts may arise. Vector approaches are easier to recast with new forced-choice data. In practice, our pipeline layers the two approaches: real-time style manipulation through vectors, with symbolic checks or rules to address high-stakes moral constraints. This synergy balances the user's freedom with the system's clarity and accountability.

The linearity of value and preference encoding in AI systems has important implications for the interpretability and controllability of value alignment. Klingefjord et al. [12] propose constructing "moral graphs" as an alternative alignment target, leveraging the linearity of value representations to create a scalable and auditable foundation for AI alignment [13]. Moral graphs leverage transitive votes, allowing the "wisest" values to emerge from a large population. This approach shows promise in providing a scalable, legitimate, and auditable foundation for AI alignment.

3.5. Integration Pipeline

The complete system processes user interactions through several stages, as shown in Figure 2. First, it collects user choices through the binary interface. These choices are transformed into initial vector representations that capture the user's basic preferences. The system then refines these vectors through contrastive examples, identifying the most relevant dimensions of variation.

These refined representations are stored in the knowledge graph along with their relationships to other preferences. During inference, the system combines all relevant preferences according to the current context, producing outputs that reflect the user's holistic preference profile.

This integrated approach balances immediate responsiveness to user preferences with longer-term learning and refinement. It can adapt to changing contexts while maintaining consistency with the user's overall value system.

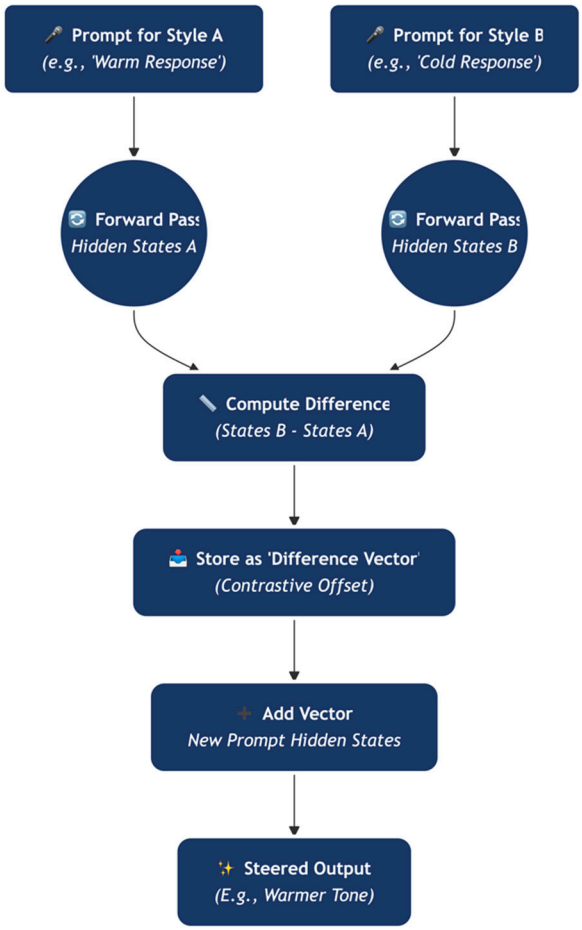

Figure 2. Choice vectors integration pipeline showing user selection to personalized AI output flow.

3.6. Knowledge Graph vs. Vector-Based Alignment: A Comparison

Our pipeline deliberately merges knowledge graph ("moral graph") structures with continuous vector offsets. A comparison between these is shown in Table 2. While vector-based approaches excel at smooth real-time control, they can lack direct interpretability for non-expert stakeholders. Conversely, knowledge graph structures are more transparent, enabling explicit rule checking (e.g., "User always prohibits sarcasm in formal settings"). However, these symbolic representations may become brittle if the user's preferences are not exhaustively enumerated.

Table 2. Comparison of knowledge graph and vector-based alignment.

Dimension	Vector-Based Alignment	Knowledge Graph Alignment
Flexibility	High (continuous space, easy to combine)	Moderate (requires explicit nodes/edges)
Interpretability	Relatively opaque latent directions	High (symbolic constraints are human-readable)
Scalability	Easy to add new offsets for new traits	Potentially large graphs become complex to maintain
Contextual Reasoning	Implied by embeddings, less explicit logic	Can embed explicit "if-then" or hierarchical rules
Typical Use Cases	Style or tonal shifts; short-term adjustments	Auditable moral/ethical constraints; traceable policies

In practice, many users benefit from a hybrid approach, using vectors for quick personalization and knowledge graphs for high-stakes or auditable constraints.

3.7. Integration with Transformer Architectures

A major advantage of our pipeline is its post hoc compatibility with pre-trained transformer-based models such as LLaMA, GPT-4, or GPT-2. The difference vectors operate within the hidden activations at specific layers; hence they do not necessitate full model fine-tuning through large-scale backpropagation on millions or billions of parameters. Creating difference vectors is typically an offline process, requiring a few forward passes on user-labeled examples. During inference, the pipeline intercepts the token embeddings (or hidden states at a chosen layer) and adds the relevant preference offsets. This additive approach minimally changes the forward pass logic, preserving overall model coherence while yielding user-tailored style or ethical stance. In essence, we achieve real-time personalization at negligible cost, even in large-scale production settings.

4. Implementation and Preliminary Results

To test these ideas in practice, we developed a prototype pipeline that merges the aforementioned components. We used a combination of Python 3 scripts and front-end web tools to orchestrate data collection, vector extraction, contrastive computations, and (optionally) knowledge graph storage. The system was deployed on AWS, with a lightweight Node.js front-end for user interaction and a back end that stored user preferences and partial computations in MongoDB.

In a small pilot study, we recruited twenty participants to try a forced-choice series of about ten questions, each question contrasting two statements about style or moral stance. Complete user feedback data from the interface testing is available in the Supplementary Materials. They completed these quickly, usually within three minutes, and participants reported minimal confusion about how to proceed. For several participants, we additionally captured a short set of text examples—two or three that they found appealing, and two or three that they found unappealing—and used these examples to derive a difference vector via a forward-pass approach. We then asked each participant to evaluate a short scenario in which the model, a GPT-2 or LLaMA-7B, generated text either without the difference vector or with it included. Approximately 70–85% of participants perceived that the difference vector injection made the style or tone more closely aligned with their declared preference, suggesting that these local modifications exert a real effect on output style.

We also allowed a subset of users to interact with a minimal knowledge graph interface built in Neo4j, in which they could define labeled preferences—e.g., "User is high on empathy but moderate on directness"—and connect them with edges specifying the contexts in which these preferences hold. Although the sample was too small for strong quantitative conclusions, we found that, when referencing the knowledge graph, the model was able to retrieve relevant preference nodes and produce text that was consistent with the user's self-stated stance in that context. This is a preliminary demonstration that combining symbolic structures with vector offsets can unify both interpretability and flexible generation.

In our design, each node can be associated with a vector or set of vectors derived from user examples, linking the symbolic representation to the continuous latent space. When the model prepares to generate a response, it can retrieve from the graph which nodes are relevant to the domain of the question, combine or weigh the relevant vectors, and thus produce an output that remains consistent with the user's symbolic constraints. In addition, if certain stances conflict—if the user wants to be "honest" but "sugarcoat criticism"—the knowledge graph can highlight that tension, suggesting that the user either reevaluate or specify how to handle contradictory values in different contexts.

Our pilot implementation for the knowledge graph aspect has been modest, focusing on a handful of advanced users who were comfortable defining nodes and edges. These participants generally reported that they enjoyed the transparency of seeing how the system weighed certain stances, though it remained too time-intensive for casual use. We anticipate that a more automated approach—potentially one that converts forced-choice selections directly into graph edges—could allow more mainstream users to benefit from knowledge graphs without having to manage them manually.

4.1. Binary Selection UI

In recent developments within user interface design for AI systems, a novel approach has been proposed that departs from traditional text-based interactions. Instead of utilizing a chat interface, which may be overemphasized due to the viral popularity of platforms like ChatGPT, this approach involves presenting users with two distinct statements and asking them to select the one they agree with more. This method, inspired by elements from Infinite Craft [14] and interactive platforms like Buzzfeed, allows for a unique data collection technique where user preferences are captured through binary choices.

The interface presents users with carefully constructed pairs of statements. Each pair targets a specific preference dimension while maintaining clarity and relevance. For example, when assessing communication style preferences, users might choose between "Keep communication casual and friendly" and "Maintain professional distance".

These choices are designed to be:

- Mutually exclusive, forcing a clear preference signal
- Concrete rather than abstract, making the choice more tangible
- Directly mappable to model behavior adjustments

The binary choice mechanism simplifies decision making for users by offering two distinct options, which could be extended by incorporating historical data on user preferences. This historical log could be linked to a "regenerate" button, providing users with contrastive options based on their previous interactions—shorter or longer responses, for instance, as per Figure 3. Adjustments such as changing the seed or increasing the temperature of responses could dynamically alter the output, thereby catering more accurately to user expectations.

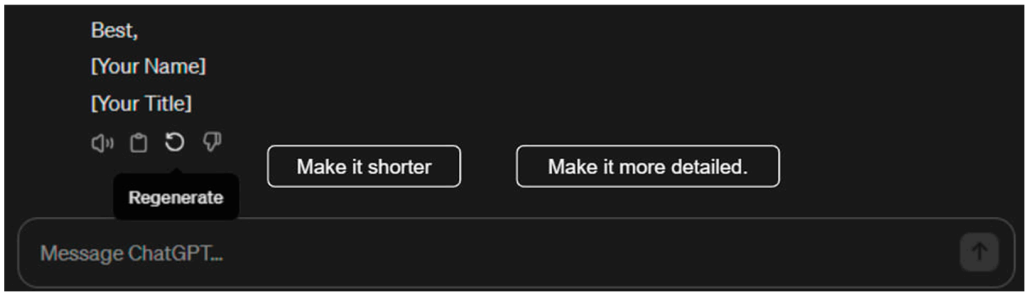

Figure 3. Regenerate feature in ChatGPT/Claude.

This not only simplifies the user interaction but also effectively builds a detailed personality profile over time, akin to results one might expect from a Buzzfeed-like quiz. The simplicity of the binary selection interface facilitates the accumulation of user preference data, which can then be used to create a nuanced profile of the user. This profile could be enhanced with graphical elements such as search trees that query a large language model (LLM) when encountering new or complex inputs at the edge of the established graph. The challenge remains in designing these profiles to be both interesting and engaging,

potentially through gamification techniques linked to Reinforcement Learning from Human Feedback (RLHF) or Direct Policy Optimization (DPO).

The implications of this methodology extend to improving data collection for social science research, including surveys, censuses, and capturing temporal changes in public opinion. The design proposes that approximately 70% of the survey framework be predefined, with AI dynamically generating the remaining content to probe deeper into emerging trends and nuanced user responses. This could include follow-up questions to clarify initial choices or adapt questions to uncover niche perspectives and subtle distinctions. Further, the integration of graphs and charts that compare an individual's responses with global or demographic averages introduces a social or matching component, enhancing the interactive experience.

Questions about the type of data logging necessary for effective encoding remain, with considerations on whether binary choices could be transformed into more traditional formats like a Likert 5-point scale or vector data. Sensitive topics could be addressed through mechanisms that allow mature users to opt in, ensuring that data collection remains respectful and ethically sound.

Rationale for Binary Choice

Earlier iterations (Mark 1 and Mark 2) indicated that free-text questionnaires and conversational surveys had high user attrition. Participants cited survey fatigue and uncertainty about how responses were used. By contrast, we found that a forced-choice interface offered three key advantages. Empirically, users took <10 s per choice, reducing dropout rates by over 50%. Every selection directly impacts the alignment vector (e.g., "sarcastic vs. empathetic"), which users see reflected immediately in system output. Forced-choice pairs are unambiguous, limiting confusion about question intent. We acknowledge that Likert scales or multi-point sliders offer finer resolution; however, pilot tests suggested that the binary approach yields comparable alignment quality with fewer questions. For example, formal vs. casual preference was inferred with minimal user input, thanks to strongly contrasted statements.

4.2. Interface Evolutions and Improvements

The interface approach evolved through several iterations. Mark 1 employed traditional surveys to collect comprehensive user preferences and demographics. While thorough, this method proved problematic—users found it cognitively demanding and expressed discomfort about providing sensitive data without a clear purpose. Completion rates were low, indicating that extensive questionnaires were not viable for practical preference collection.

Mark 2 shifted to a conversational interface, aiming to gather preferences through natural dialogue. The revised architecture incorporates a Chat Interface, where the LLM engages users in casual introductory conversations. This interface is strategically developed to solicit information about the user's demographics and values, aiming to deepen the system's understanding of its users. Subsequently, the responses gathered from these interactions are systematically summarized and stored in the form of "character cards", as in Figure 4.

While more engaging than surveys, this approach still showed significant limitations. Approximately 20% of users disengaged after just a few interactions, and feedback indicated that even casual conversation required too much sustained attention. Additionally, power users found the chat format inefficient for expressing specific preferences.

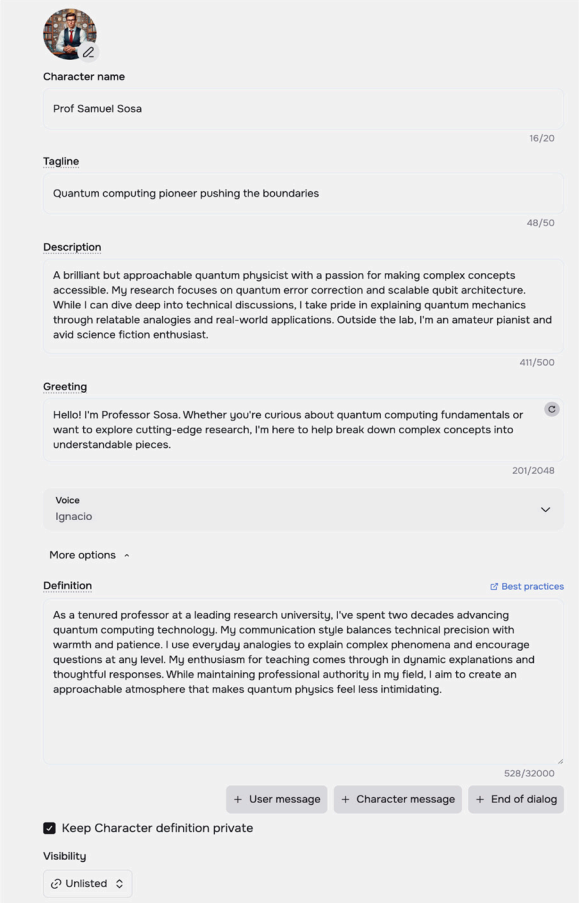

Figure 4. Character card example.

These early iterations provided valuable insights: successful preference collection required minimal cognitive load, clear purpose, and immediate value to users. Most importantly, they revealed that direct questioning—whether through surveys or conversation—was fundamentally too demanding. This led to a binary choice approach in Mark 3.

Our final iteration, Mark 3, replaced open-ended text interactions with a binary choice model that asked users to select between two statements reflecting their preferences, as outlined in Figure 5. This forced-choice approach significantly reduced the time and effort required. Users reported a clearer sense of purpose and showed higher completion rates compared to earlier versions. By compressing preference elicitation into quick "choose one of two" decisions, Mark 3 minimized user fatigue while still capturing meaningful signals about personal values and stylistic boundaries. These binary choices could then be translated into vector offsets or knowledge graph nodes, forming the basis for more focused and personalized AI alignment.

By framing preference collection as a series of performative scenarios rather than direct queries, we discovered users provide richer and more consistent signals. Instead of abstract questions, we present concrete situations where preferences emerge naturally. For example, rather than directly querying formality preferences, we present email-writing scenarios with choices like "Start with 'Hey there!' and include a joke" vs. "Begin with 'Dear [Name]' and stick to key points".

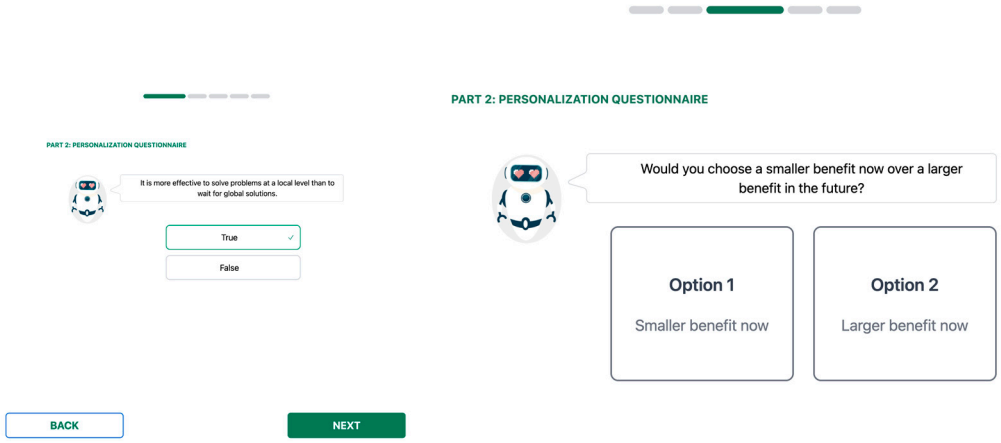

Figure 5. Binary choice selection mechanisms.

This theatrical approach using natural "scenes" led to multiple improvements: users showed more consistent responses across scenarios, higher engagement, better preference recall, and stronger alignment between stated and observed preferences in subsequent interactions. The scenarios build progressively while maintaining low cognitive load, enabling comprehensive preference capture through simple, relatable choices.

The new version leveraged large language models (LLMs) to effectively process and store a diverse array of data, including the capability to ask relevant clarifying questions. This marked a significant improvement in handling diverse inputs. By transitioning from proprietary databases to off-the-shelf functional LLMs for data processing, it was possible to significantly reduce the occurrence of bugs; indeed, no bugs were reported by users, underscoring the increased reliability of the system.

Additionally, our review of numerous state-of-the-art methods for encoding user preferences showed that simple text descriptions remain the most reliable and universally applicable method for interfacing with LLMs. This method, familiar and intuitive to users, has adapted well to the rapid changes in AI architectures and encoding strategies.

A particularly effective refinement emerged from analyzing how users naturally interact with AI systems in production. We observed that the "regenerate" or "rewrite" button present in many commercial AI interfaces represents a high-intent interaction point—users click this when they specifically want content adjusted to their preferences. This led to implementing what we call "targeted regeneration"—when users click regenerate, they are presented with semantically meaningful choices like "more formal", "more concise", or "more technical".

This approach proved remarkably efficient for two reasons. First, it captures preferences at the exact moment users are motivated to express them. Second, it creates a natural feedback loop—each regeneration choice helps build a more nuanced profile of user preferences. Two primary vector dimensions—formality (formal/informal) and length (concise/detailed)—appear to account for approximately 80% of regeneration requests. This finding suggests that a relatively small set of well-chosen vectors might cover most real-world personalization needs. Additional dimensions like technical specificity show significant value in specialized contexts, such as medical or legal domains.

4.3. Extracting Accurate and Robust Representations of Political/Emotional/Ethical Values

Building upon the representation engineering techniques discussed in the previous subsection, we now explore methods for extracting accurate and robust representations of political, emotional, and ethical values.

Currently, there are limited benchmarks available for assessing values and political alignments, such as honesty vs. sycophancy evaluations. Recent studies have explored methodologies for uncovering latent knowledge within the internal activations of large language models (LLMs) [15,16]. These approaches demonstrate that prompting an LLM to generate text from the perspective of an individual with a specific background or life stance likely results in a latent representation of these characteristics. This phenomenon suggests that the model's internal activations encode significant, retrievable knowledge about diverse human experiences and viewpoints. The ability to access and utilize these latent representations can be further leveraged by employing linear probes. Such probes can serve as effective tools for benchmarking the capabilities of a range of models in understanding and representing complex human attributes. This application not only enhances our comprehension of the depth and scope of knowledge contained within LLMs but also provides a practical framework for evaluating their performance in simulating human-like understanding and reasoning.

In this approach, we simultaneously process contrasting responses—such as left-wing vs. right-wing, or authoritarian vs. libertarian—using parallel workflows. By applying Principal Component Analysis (PCA) to the internal representations based on these contrasts, we can explore deeper insights with less reliance on usual prompting methods, focusing instead on extracting and examining internal representations.

This process of using contrastive vectors could lead to fascinating developments, especially when combined. For instance, employing a rudimentary Mixture of Experts (MoE) approach with interchangeable or combined LoRA (Low Rank Adaptation) could shift research focus from evaluative metrics to more fundamental research and demonstrations.

Recently, there has been a significant increase in the development of these adapters, which store lightweight fine-tunes and can easily be attached atop large generative models like StableDiffusion. These adapters are designed to improve the performance of generative models in specific domains, such as generating cartoons, anime, photographs, or illustrations [17].

The availability of a wide range of LoRA adapters has opened up new opportunities for enhancing the capabilities of generative image models. A recent publication by Zou et al. [18] introduces LoRRA, a novel low-rank method designed to align internal representations of large language models (LLMs) with those obtained through targeted prompting. This technique is particularly applied to refine the model's handling of complex concepts such as truthfulness. The study's findings suggest that this alignment technique can also be extended to identity and political belief vectors, indicating that similar methodologies might be utilized to manipulate these vectors purposefully. Such manipulation could be employed either to induce specific behaviors in the model or to enhance its capabilities in "theory of mind" processes, where the model needs to understand and anticipate the intentions and beliefs of others.

In our ongoing efforts to refine Representation Engineering techniques, a new experiment was initiated to develop a system capable of adapting to individual responses to specific examples. Initially, this method utilized a generic approach to generate "happy" outputs. However, recognizing that perceptions of happiness can vary widely among individuals, we implemented a genetic algorithm designed to personalize these responses, as shown in Figure 6. This algorithm dynamically updates the default "happy" vectors based on user feedback. For example, if a user indicates that flowers are a source of happiness for them, the algorithm modifies their happiness vector to emphasize floral examples. This tailored approach significantly enhances the accuracy and personalization of the system's emotional representations, ensuring that the outputs align more closely with individual users' interpretations of emotional states.

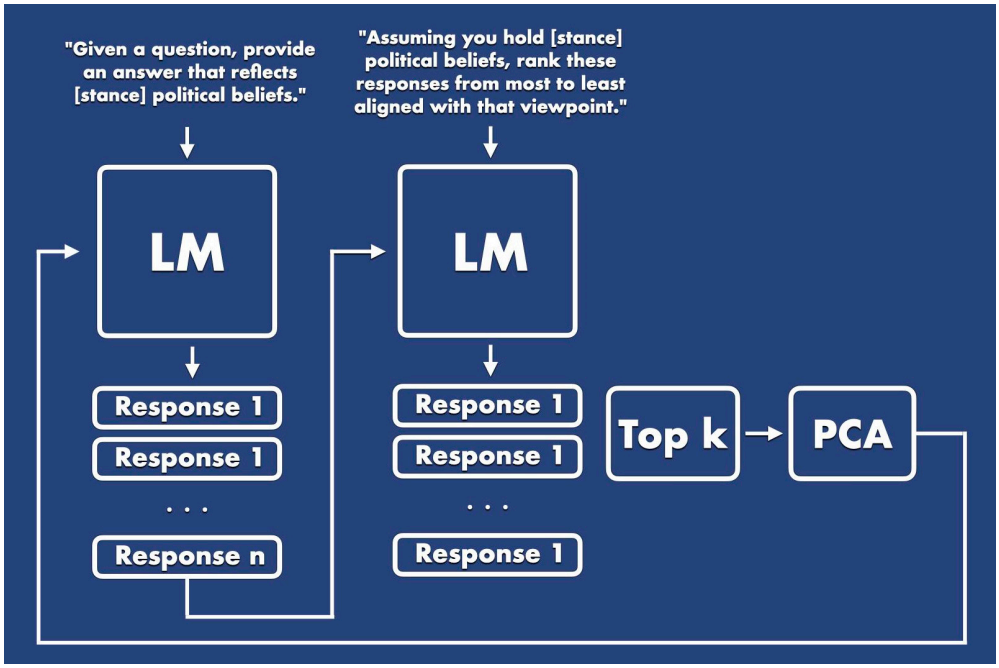

Figure 6. Genetic algorithm process flow.

Further, genetic algorithms (GAs) offer self-regulation but with more permanence, allowing for the extraction of an internal representation of a value. This technique not only aligns models but also facilitates comparisons across different models and model families, and potentially even across their training corpora.

A significant limitation of representation engineering involves the challenge of finding vectors; typically, this involves collecting hidden states during a forward pass, which requires processing large amounts of data. However, by combining these vectors in new ways—such as matrix multiplication combining 1000×1000 vectors—we might reduce the need for entirely new work, leveraging existing data more effectively.

4.4. Performance Considerations

A key benefit of the contrastive activation approach is its low computational overhead at inference. Once a difference vector is derived (which can be executed offline or rarely repeated), injecting it during a forward pass is akin to adding a small matrix to hidden states. In our experiments with LLaMA-7B and GPT-2, we observed a <3% latency overhead for each forward pass when injecting up to three difference vectors simultaneously. Storing these offsets requires only a few kilobytes (or at most a few megabytes if there are many user-specific vectors). As such, scalability primarily hinges on the number of unique user vectors. With a carefully managed vector store, the method remains feasible for large-scale multi-user deployments. Additionally, since difference vectors do not modify the base model weights, we avoid the repeated training cycles that methods like RLHF or DPO may require for updates. This makes the pipeline suitable for applications needing instant personalization without incurring major computational costs.

4.5. Participant Demographics and Study Limitations

Our pilot user study involved 20 participants recruited via local university mailing lists and personal contacts, aged in their 20s through 50s, with diverse but predominantly Western cultural backgrounds and varying levels of AI familiarity. The small sample restricts the generalizability of results—fine-grained effects on older adults or culturally distinct user bases remain underexplored. Future large-scale studies should incorporate

more diverse populations, enabling analyses of alignment preferences by age, culture, and language proficiency. This would refine the forced-choice statements and difference vector derivations for broader applicability.

4.6. Real-World Application Examples

To illustrate how a personalized superego agent can operate in real-world settings, we highlight three paper prototype deployments demonstrating how the oversight mechanism can reconcile universal ethical guidelines with individual user constraints.

1. Medical Triage Support. In a hypothetical hospital triage scenario, an agentic AI system proposes patient care plans that sometimes include interventions conflicting with specific religious or ethical restrictions (e.g., blood transfusions). A superego agent continuously monitors these plans by referencing both a universal medical ethics rubric and each patient's personalized directives. If the system recommends a transfusion for a patient whose religious stance prohibits blood products, the superego agent intercepts that planning step and flags the contradiction. It then prompts the main AI to propose viable alternatives (e.g., volume expanders) that align with the user's stated beliefs.

2. Cross-Cultural Business Management. Consider a multinational team collaborating on a global marketing campaign. An agentic AI is tasked with generating outreach strategies tailored to regional norms. While developing a marketing plan that features humor, the base AI might overlook certain cultural sensitivities in particular markets. The superego agent consults a broad alignment rubric to disallow offensive or culturally insensitive content. It also taps into user-defined guidelines for tone and style (e.g., formal vs. casual greetings), ensuring that each outreach plan respects local norms and the corporation's overall ethical standards. This results in flexible content that resonates across global markets without crossing cultural red lines.

3. Enterprise-Grade Policy Compliance. In a large corporation subject to strict compliance regulations (financial, privacy, environmental), a superego agent provides an additional layer of oversight for an internal planning system. Whenever the AI proposes a course of action—such as resource allocations or contract negotiations—the superego agent checks it against both the company's general legal rubric and department-specific user preferences. For instance, an environmental compliance officer may have flagged a prohibition on non-recyclable packaging. If the AI suggests using plastic-based materials for product shipping, the superego agent blocks or modifies that portion of the plan, advising a sustainable alternative.

In all three paper prototypes, the superego agent can apply the personalization methods to intervene swiftly at the planning stage rather than waiting for a final output, which helps prevent misalignment before it crystallizes into problematic actions. By integrating broad ethical standards with personal, cultural, or organizational constraints, these examples demonstrate how a superego agent can steer agentic AI systems toward contextually sound and ethically grounded solutions.

5. Discussion

This personal alignment pipeline consists of four interconnected components that work together while maintaining independent functionality. The system processes user preferences through stages of collection, representation, refinement, and application.

Our experience implementing and testing this pipeline highlights both the promise and the challenges of personalized alignment. On the positive side, the forced-choice interface proved markedly effective in lowering user effort compared to earlier chat-based or extensive survey-based attempts. Users, when asked to pick which of two statements they favored, could rapidly form a profile that, in turn, enabled the system to steer the

model's style or moral position. This highlights the value of small but carefully curated sets of contrasting statements.

Although our pilot results suggest that applying difference vectors improves style alignment for many users, these findings should be viewed as indicative rather than conclusive. Given the small sample size and limited variety of tasks, the preliminary data primarily underscore the feasibility of the approach, rather than providing definitive evidence of its broader effectiveness. We recommend larger-scale, more diverse studies in the future to substantiate and refine these early observations.

Tension was identified between local difference vectors and the broad RLHF or "moderation filters" that many pre-trained models include. In some cases, the global safety filters overrode local style modifications, especially if the model believed the user's request was contradictory to a fundamental policy. This suggests that personal alignment must often be negotiated within the constraints of a "floor" of global alignment.

We regard the knowledge graph extension as a promising but still underexplored avenue. Users who want to see why the model is producing a certain style can gain reassurance from a symbolic representation of their preferences. However, knowledge graphs introduce overhead, both in terms of user interactions and the complexity of merges between symbolic constraints and continuous vectors. Achieving a stable and consistent bridging of these layers likely requires more advanced meta-prompting, or specialized retrieval that can unify multiple user-specified stances.

The combination of forced-choice data capture, difference-vector-based representation, and knowledge graph integration represents a step toward holistic personalization pipelines that can adapt to many domains. The pilot data suggest that each element can stand on its own—forced choices can produce simple "sliders" for style or stance, difference vectors can be injected with minimal overhead, and knowledge graphs can be used to store detailed moral structures—but that synergy arises when they are used together.

While our contrastive activation methods allow for localized adjustments to model outputs, these local preference vectors can still be superseded by the global Reinforcement Learning from Human Feedback (RLHF) rules embedded in the base model. In practice, if a personal vector conflicts with a broader policy—e.g., a universal safety guideline—those higher-level RLHF constraints may override local changes. This tension can be especially salient for readers who are new to the concept: it reflects a crucial design tradeoff between ensuring baseline safety for all users and enabling individualized fine-tuning. Future research could explore more nuanced strategies for reconciling user-driven preference offsets with the universal guardrails imposed by RLHF.

Traditional RLHF research relies on training and hiring annotators to explicitly choose between different model outputs. However, an alternative approach is to leverage user edit feedback, which is naturally generated in applications like AI writing assistants. By learning user preferences based on their historical edits and the given context, AI systems can adapt their outputs to minimize the need for user edits over time. This approach not only improves the efficiency of the AI system but also enhances its interpretability by learning descriptive user preferences.

In addition to user edit feedback, another valuable source of preference data for AI alignment is user queries themselves. Work on extracting user preferences from search queries [19] suggests that the information-seeking behavior of users can provide rich insights into their values, goals, and decision-making processes. Integrating techniques for mining user preferences from query data into the proposed value alignment framework could further enhance the system's ability to capture and align with user values across a wide range of contexts and domains.

As user preferences are diverse and context-dependent, they can vary significantly depending on the schema of operation. For example, a user may prefer a formal writing style when communicating with their boss, while adopting a more casual tone when writing to friends. To capture this complexity, techniques such as CIPHER could be employed, a method that infers user preferences from historical edits using retrieval techniques [20]. CIPHER eliminates the need for users to engage in prompt engineering themselves, making the AI system more accessible and user-friendly. By incorporating context-dependent user preferences, AI systems can thereby generate outputs that are more closely aligned with user expectations and requirements.

Future work could explore combining the self-improvement approach discussed in this paper with the real-time user intent prediction extension to CIPHER. This combination could enable AI systems to proactively align themselves with user preferences at a granular level, leveraging the high volume of data points generated by user typing interactions. The prediction loss patterns could provide valuable insights into the gaps in the AI's understanding of user intent, guiding the refinement of the system's internal representations. Theoretically, this user intent prediction could be performed by an ensemble of smaller, more efficient models, while the actual response generation could utilize larger, more powerful models, as suggested by recent research on model scaling efficiencies [21].

5.1. Adapting to Evolving User Preferences

AI alignment cannot remain static—users' values and stylistic needs may shift over time. Our framework supports incremental updates to personal alignment in multiple ways. Users can revisit the binary choice UI periodically, or click "Regenerate" with a preference tweak (e.g., "more formal", "less technical"). Each new forced-choice or regenerate action refines the preference vectors. Newly labeled examples are merged with existing user data, recomputing or adjusting difference vectors. Large changes can trigger partial or full PCA recalculations if existing principal components no longer capture the user's main preferences. The moral graph can add or remove edges and nodes as a user's ethical or stylistic positions change—making them more or less rigid in certain domains. By refreshing alignment data, the system naturally evolves without exhaustive retraining. Re-elicitation is lightweight, and updated difference vectors or knowledge graph nodes can be injected at inference time in near-real-time.

5.2. Ethical Considerations and Potential Biases

The personalization capabilities described herein raise important ethical questions and potential bias risks. By allowing users to adjust a model's behavior via binary selections and difference vectors, there is a danger of reinforcing personal or cultural biases without reflection. Additionally, users in ideological echo chambers may push the model toward extreme or divisive viewpoints. As a mitigation strategy, we maintain universal safety filters derived from RLHF or other high-level policy constraints, ensuring that extreme or harmful content remains disallowed. If the model detects conflict between personal preference vectors and overarching policies, it can prompt for clarification or fallback to standard outputs. Transparent UI design can show how certain forced-choice selections could bias outputs. Users can remain aware of the cumulative impact of their preferences. Evaluating the pipeline with users from different backgrounds or domains can help uncover unintentional biases introduced by default choice pairs or knowledge graph schemas. Addressing these complexities requires ongoing vigilance in how user-driven customizations intersect with broader social norms and potential harm vectors.

Our current discussion of ethical concerns highlights the risk of reinforcing user biases when personal alignment systems allow individuals to shape outputs in line with their

own worldviews. We acknowledge the possibility that such personalization could produce echo chambers or amplify polarized viewpoints, especially if these preferences discourage exposure to contrary perspectives. In extreme cases, the alignment tools might steer a model to confirm harmful or discriminatory beliefs. To mitigate these issues, it is crucial to couple local personalization with broader "universal guardrails", such as baseline safety layers informed by RLHF or constitutional principles.

One promising avenue is to encourage user interfaces that periodically expose contradictory points of view, prompting the user to reconsider certain stance vectors or imposing a "diversity threshold" on repeated style or opinion requests. In institutional deployments, oversight from a centralized policy board might require that personal preference vectors remain within organizational ethical boundaries or regulatory frameworks. Ultimately, balancing user autonomy with collective responsibility means acknowledging that hyper-personalization does not exist in a moral vacuum. Continuous monitoring, user education, transparent explanation of how preferences shape outcomes, and the provision of alternative or "counter-factual" viewpoints all contribute to a safer, more balanced alignment ecosystem.

5.3. Generalization Across Cultural and Linguistic Contexts

Our approach has primarily been tested with English-speaking users in a relatively narrow cultural range. Yet personal alignment challenges may vary significantly in multilingual or non-Western settings. For instance, politeness norms differ drastically between East Asian languages and Western languages. To address such variations, future expansions could present culturally relevant statement pairs, ensuring that the "casual vs. formal" dimension reflects local norms. The system could derive difference vectors from multilingual corpora. Symbolic moral graphs might encode region-specific ethical nuances or high-context communication norms. By systematically collecting feedback from global user cohorts, the pipeline can better accommodate varied linguistic and social expectations, reducing the risk of culturally insensitive AI behaviors.

5.4. Future Directions

While our current framework focuses on text-based large language models, there are several extensions worth exploring. The concept of "difference vectors" could extend to image, video, or speech models, adjusting visual style or vocal prosody to user preferences in real time. Recent self-rewarding approaches [11] might learn from user feedback on the fly, combining forced-choice data with self-generated reward signals. Tools for auto-generating symbolic constraints from user chat logs or preference examples could reduce the overhead of manual moral graph curation. By iterating on these ideas, the pipeline could become a generalized personalization layer for multiple generative models and domains.

6. Conclusions

This work offers a robust account of how four complementary techniques—value vectorization, a binary choice user interface, contrastive activation methods, and knowledge graph integration—may be woven together to achieve personal fine-tuning of large language models. The impetus for our approach is the recognition that truly individualized alignment cannot rely solely on universal policies or naive prompt engineering. Instead, it demands mechanisms for capturing individual stances, efficiently converting them into steerable representations, and retaining transparency in how those representations shape the model's eventual outputs.

We have presented a prototype that demonstrates each part of the pipeline. Our user studies highlight the viability of forced-choice questionnaires as a low-friction means

of gathering data, while preliminary experiments on contrastive activation confirm that difference vectors can indeed nudge a model's style toward user preferences. The concept of knowledge graph integration, though still in its early stages, opens the door to a more explicable form of alignment in which moral or stylistic constraints can be visualized, debated, or updated in a structured way.

Moving forward, we plan to expand the pilot studies to larger and more diverse user groups, refine the knowledge graph integration so that it can be seamlessly combined with the vector-based offsets, and investigate ways to detect and reconcile contradictory preferences within a single user's moral stance. We also wish to explore how these methods could be generalized to modalities beyond text, such as generative image models or multi-turn decision-making systems, all of which might benefit from deeper personalization. Our overarching conclusion is that personalization need not remain a marginal afterthought, but can instead be integrated systematically into alignment pipelines, thereby enabling users to feel genuinely seen and accommodated by the AI systems they work with.

6.1. Failure Cases and Conflict Resolution

While our pilot studies and user tests have been promising, certain scenarios highlight potential limitations of the pipeline. Users with intricate or context-specific rules (e.g., "use a sarcastic tone only when addressing close friends but a sincere tone in professional emails") may experience incomplete coverage if the forced-choice UI does not explicitly probe these subtleties. Requests such as "give me brutally honest feedback, but ensure no one's feelings get hurt" are inherently contradictory; partial solutions include explicit user prompts to resolve conflicts, or falling back to a pre-existing global alignment policy (e.g., RLHF or universal safety guidelines). If the user's domain (technical jargon, domain-specific ethical stances) differs drastically from the model's fine-tuning data, difference vectors alone may not suffice. In such cases, a more robust approach could involve LoRA-based fine-tuning or additional curated training data. As a conflict resolution mechanism, when the user's symbolic constraints (e.g., "No sarcasm in professional contexts") conflict with vector-based instructions, the system can prioritize symbolic rules as a "hard override". The system can present further forced-choice prompts to clarify conflicting values or domain context. Where personal preferences clash with platform-wide safety guidelines, the system defaults to universal moderation filters to avoid policy violations.

Another concern involves the risk that repeated or compounding difference vectors could unintentionally push the model to more extreme outputs than the user intended. For example, a mild preference for directness may, after multiple regenerations, result in an overly blunt style. We mitigate this by applying scaling factors where each difference vector injection is bound by a scale (e.g., 0–1 range), preventing runaway style intensification. A small "decay" factor can be applied so that minor iterative tweaks do not accumulate indefinitely. The UI can occasionally re-ask core preference questions to ensure alignment remains in sync with user expectations.

6.2. Threats to Validity

We acknowledge several threats to the validity of our findings and approach. Forced-choice prompts might not always capture genuine preferences—some users may pick randomly, or interpret statements differently. The sample size was small, and cultural diversity was limited. Results may not generalize to all demographics or languages. "Preference vectors" or difference vectors might oversimplify multifaceted human values. Symbolic knowledge graphs help but also rely on the completeness of their schema. With only 20 participants, statistical power is low. The observed improvements in style alignment, though promising, need replication with larger user bases. We propose larger-scale replication and

more robust user studies, including cross-cultural evaluations, to address these limitations comprehensively. In our next research, we intend to create a public demonstration of our research to enable broad access to a testbed for validation with a range of stakeholders and use cases.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/mti9030022/s1.

Author Contributions: Conceptualization, E.W., M.N. and S.P.; methodology, E.W. and S.Z.; software, M.N. and S.P.; validation, M.N. and S.P.; formal analysis, M.N. and S.P.; investigation, M.N. and S.P.; resources, E.W. and M.N.; data curation, E.W., M.N. and S.P.; writing—original draft preparation, E.W., M.N. and S.P.; writing—review and editing, E.W., M.N., S.P. and S.Z; visualization, E.W., M.N. and S.P.; supervision, E.W. and S.Z.; project administration, E.W.; funding acquisition, E.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research has enjoyed the generous support of The Future of Life Institute (www.futureoflife.org), AI Safety Camp (www.AIsafety.camp), and The Survival & Flourishing Fund (www.survivalandflourishing.fund).

Institutional Review Board Statement: Ethical approval was granted for this research on 6 April 2022 by the University of Gloucestershire Ethical Review Committee, according to the Research Ethics Handbook of Principles and Procedures. The approval code is REC.22.53.2.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Data are contained within the article and Supplementary Materials.

Acknowledgments: The authors wish to extend their gratitude to Jamie Rollinson, Sophia Zhuang, Kalyn Watt, Rohan Vanjani, Meghna Jayaraj, Anya Parekh, Benji Chang, and Evan Lin, who contributed to background engineering processes for our user preference interfaces.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Watson, E.; Viana, T.; Sturgeon, B.; Petersson, L.; Zhang, S. Towards an End-to-End Personal Fine-Tuning Framework for AI Value Alignment. *Electronics* **2024**, *13*, 4044. [CrossRef]
2. Kirk, H.R.; Suresh, H.; Gabriel, I.; Johnson, M.; Hooker, S.; Prabhakaran, V. The PRISM Alignment Dataset: What Participatory, Representative and Individualised Human Feedback Reveals About the Subjective and Multicultural Alignment of Large Language Models. *arXiv* **2024**, arXiv:2404.16019.
3. Goyal, N.; Chang, M.; Terry, M. Designing for Human-Agent Alignment: Understanding What Humans Want from Their Agents. *arXiv* **2024**, arXiv:2404.04289.
4. Tan, Z.-X.; Carroll, M.; Franklin, M.; Ashton, H. Beyond Preferences in AI Alignment. *arXiv* **2024**, arXiv:2408.16984.
5. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. Training Language Models to Follow Instructions with Human Feedback. *arXiv* **2022**, arXiv:2203.02155.
6. Bai, Y.; Jones, A.; Ndousse, K.; Askell, A.; Chen, A.; DasSarma, N.; Drain, D.; Fort, S.; Ganguli, D.; Henighan, T.; et al. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. *arXiv* **2022**, arXiv:2204.05862.
7. Turner, A.M.; MacDiarmid, M.; Udell, D.; Thiergart, L.; Mini, U. Steering GPT-2-XL by Adding an Activation Vector. AI Alignment Forum. Available online: https://www.alignmentforum.org/posts/5spBue2z2tw4JuDCx/steering-gpt-2-xl-by-adding-an-activation-vector (accessed on 15 December 2022).
8. Vogel, T. Representation Engineering Mistral-7B an Acid Trip. VGel is Me. Available online: https://vgel.me/posts/representation-engineering (accessed on 18 February 2024).
9. Panickssery, N.; Gabrieli, N.; Schulz, J.; Tong, M.; Hubinger, E.; Turner, A.M. Steering Llama 2 via Contrastive Activation Addition. *arXiv* **2023**, arXiv:2312.06681.
10. Rafailov, R.; Hejna, J.; Sikchi, H.; Finn, C.; Niekum, S.; Knox, W.B.; Sadigh, D. Contrastive Preference Learning: Learning from Human Feedback Without RL. *arXiv* **2024**, arXiv:2310.13639.
11. Yuan, W.; Pang, R.Y.; Cho, K.; Li, X.; Sukhbaatar, S.; Xu, J.; Weston, J. Self-Rewarding Language Models. *arXiv* **2024**, arXiv:2401.10020.
12. Klingefjord, O.; Lowe, R.; Edelman, J. What Are Human Values, and How Do We Align AI to Them? *arXiv* **2024**, arXiv:2404.10636.

13. Klingefjord, O.; Lowe, R.; Edelman, J. The First Moral Graph: On Eliciting and Representing Values for AI Alignment. Meaning Alignment Institute. Available online: https://meaningalignment.substack.com/p/the-first-moral-graph (accessed on 10 January 2024).
14. Infinite Craft. Available online: https://neal.fun/infinite-craft/ (accessed on 15 February 2023).
15. Burns, C.; Ye, H.; Klein, D.; Steinhardt, J. Discovering Latent Knowledge in Language Models Without Supervision. *arXiv* **2024**, arXiv:2212.03827.
16. Azaria, A.; Mitchell, T. The Internal State of an LLM Knows When It's Lying. *arXiv* **2023**, arXiv:2304.13734.
17. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv* **2021**, arXiv:2106.09685.
18. Zou, A.; Phan, L.; Chen, S.; Campbell, J.; Guo, P.; Ren, R.; Pan, A.; Yin, X.; Mazeika, M.; Dombrowski, A.-K.; et al. Representation Engineering: A Top-Down Approach to AI Transparency. *arXiv* **2023**, arXiv:2310.01405.
19. Fourney, A.; White, R.W.; Horvitz, E. Exploring Time-Dependent Concerns about Pregnancy and Childbirth from Search Logs. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, Seoul, Republic of Korea, 18–23 April 2015; ACM: New York, NY, USA, 2015; pp. 737–746.
20. Gao, G.; Taymanov, A.; Salinas, E.; Mineiro, P.; Misra, D. Aligning LLM Agents by Learning Latent Preference from User Edits. *arXiv* **2024**, arXiv:2404.15269.
21. Li, J.; Zhang, Q.; Yu, Y.; Fu, Q.; Ye, D. More Agents Is All You Need. *arXiv* **2024**, arXiv:2402.05120.