Bridging Personalization and Control in Scientific Personalized Search

Sheshera Mysore
University of Massachusetts
Amherst, USA
smysore@cs.umass.edu

Garima Dhanania
University of Massachusetts
Amherst, USA
garimadhanania@gmail.com

Kishor Patil
Lowe's Companies, Inc
Bangalore, India
kishor.patil@lowes.com

Surya Kallumadi
Coursera
Mountain View, USA
surya@ksu.edu

Andrew McCallum
University of Massachusetts
Amherst, USA
mccallum@cs.umass.edu

Hamed Zamani
University of Massachusetts
Amherst, USA
zamani@cs.umass.edu

Abstract

Personalized search is a problem where models benefit from learning user preferences from per-user historical interaction data. The inferred preferences enable personalized ranking models to improve the relevance of documents to users. However, personalization is also seen as opaque in its use of historical interactions and is not amenable to users' control. Further, personalization limits the diversity of information users are exposed to. While search results may be automatically diversified this does little to address the lack of control over personalization. In response, we introduce a model for personalized search that enables users to control personalized rankings proactively. Our model, CtrlCE, is a novel cross-encoder model augmented with an *editable memory* built from users' historical interactions. The editable memory allows cross-encoders to be personalized efficiently and enables users to control personalized ranking. Next, because all queries do not require personalization, we introduce a *calibrated mixing model* which determines when personalization is necessary. This enables users to control personalization via their editable memory only when necessary. To thoroughly evaluate CtrlCE, we demonstrate its empirical performance in four domains of science, its ability to selectively request user control in a calibration evaluation of the mixing model, and the control provided by its editable memory in a user study.

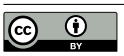
CCS Concepts

• **Information systems** → **Personalization**.

Keywords

controllable personalization; cross-encoders; calibrated retrievers

ACM Reference Format:
Sheshera Mysore, Garima Dhanania, Kishor Patil, Surya Kallumadi, Andrew McCallum, and Hamed Zamani. 2025. Bridging Personalization and Control in Scientific Personalized Search. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25), July 13–18, 2025, Padua, Italy*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3726302.3729913



SIGIR '25, Padua, Italy

ACM ISBN 979-8-4007-1592-1/2025/07
https://doi.org/10.1145/3726302.3729913

1 Introduction

Personalized search powers several industry scale search systems for products [5, 57], movies [38], jobs [24], and web-search more broadly [25]. While personalization in search systems improves the relevance of search results and increases the uptake of systems, personalized systems are commonly seen as opaque and failing to provide users with sufficient control over personalized predictions [21, 31]. Prior work has noted that personalized ranking is more likely to prevent users from seeing the breadth of information in a document collection, raising concerns of fairness [14] and hindering proactive exploration in learning-oriented applications such as education and science [43].

Such concerns about personalization have been addressed through two avenues: diversifying search results and enabling interactive control over personalized ranking. While diversification of search results is meaningful [41, 50], it does not improve user control or facilitate proactive user-driven interaction and discovery [42]. To remedy this, a small body of work has explored providing users interactive control over personalized search by rendering user representations used for personalization "editable" [1, 60]. However, this work has focused on designing visualization interfaces for user control with simpler token/entity-based user representations. While prior work on controllable personalization for *search* has been limited, a significant amount of work has explored scrutable/controllable approaches for personalized *recommendation*. This work has explored technical approaches for scrutable recommendations [6, 31] and run human-centered evaluations to show how control over personalized recommendations improved user satisfaction and trust [26, 29]. In this paper, we take inspiration from this work and extend the technical body of work on controllable personalized search.

We begin by outlining the following goals for controllable personalized search: (1) to allow control, the user representations used for personalization must be transparent to users and should enable users to express preferences through intuitive profile edits. (2) Since search can be performed without any personalization a controllable model should enable users to opt-out of personalization, supporting "no personalization" and (3) Since only some queries are likely to require personalization [2, 48] a controllable model should highlight the queries for which user profile control would be meaningful.

To fulfill these goals we introduce CtrlCE (see Figure 1), a controllable cross-encoder model personalized with an *editable memory* constructed from historical user documents. We explore two

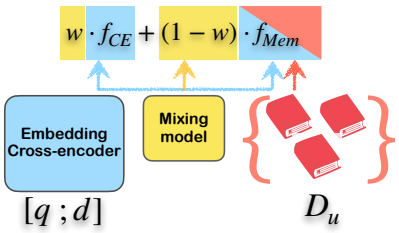


Figure 1: Our approach CtrlCE, augments a cross-encoder with an editable user profile using a calibrated mixing model. Our training procedure ensures that the mixing models score w remains proportional to the performance of f_{CE}. This ensured that it can be used for seeking edits to a user profile only when necessary.

multi-vector user memories, an item-level memory and a novel concept-based memory introduced in recent work on controllable recommendation [37]. While both representations remain transparent and editable, concept-based memories offer a richer set of edit operations than item-based memories – we detail this in §4.3. Next, to ensure that the CtrlCE cross-encoder benefits, both, from the rich query-document interaction common in cross-encoders and from interaction with the editable user memory we formulate it as a novel *embedding cross-encoder*. This diverges from standard `CLS` token-based cross-encoders and instead learns separate but contextualized query and document embeddings. Finally, we train a novel *calibrated mixing model* which intelligently combines query-document and user-document scores while also identifying the queries that are likely to benefit from user profile edits.

In experiments on datasets of personalized search from four scientific domains CtrlCE outperforms standard personalization approaches based on dense retrieval, personalized ensemble models, and non-personalized approaches spanning sparse, dense, and cross-encoder retrievers/re-rankers by 6.4-10.6% across evaluation metrics. Then we demonstrate that CtrlCE fulfills the goals of controllability: empirically demonstrating its ability to perform with "no personalization", showing in calibration evaluations that it effectively identifies queries that need personalization e.g. under-specified and exploratory queries, and showing in a user study that performance can be improved through interaction with the user profile. To the best of our knowledge CtrlCE is the first approach for controllable personalized search with language model based cross-encoders and extends an under-explored area. We release code and data at: https:/github.com/iesl/controllable-personalization-ctrlce

2 Related Work

Personalized search. Personalization has broadly been explored in search over personal document collections and referred to as "personal search" [11, 28] and search over a shared document collection [47]. Work on personal search commonly suffers from under-specified or context-dependent queries and sparse user document interactions. Therefore, prior work has focused on leveraging metadata [8] and contextual information such as time and location of queries to improve performance [40, 59] or on training schemes to learn from sparse interactions [11, 52]. On the other hand, personalized search has focused on constructing user models from users' historical interactions and using them for re-ranking documents – this is more relevant to our work.

To build user representations early work leveraged term level models [47], topic models [46], and latent representations learned through matrix factorization [12]. More recent work has learned personalized word embeddings [56], and learned user representations with RNNs [2, 3] and transformers [10]. Current work has leveraged dense retrieval models fine tuned for personalized re-ranking [61, 62]. In relying on pre-trained language models CtrlCE resembles these approaches. However, prior work leverages attention and shallow transformer layers for query-document and user-document scoring. In contrast, our work leverages a query-document cross-encoder delivering stronger performance – we show this in Section 5.2. Dai et al. [17] present an exception and leverage cross-encoders that input *all* historical context, query, and documents for personalized product search. Notably, this is only feasible with short historical texts, queries, and documents common in product search. In Section 6.1 we also show its inability to be controllable.

Controllable personalized search. While the above approaches explore personalized search, they don't consider control over personalization. The most relevant prior work is provided by Zemede and Gao [60] and Ahn et al. [1] – both of who explore visualization interfaces for interacting with term/entity-based user profiles in personalized search. While meaningful, this work does not explore performant retrieval models as ours does. Finally, recent work [33, 51] explores combining personalization and search result diversification by learning when personalization is necessary while sharing in our motivation these approaches do not enable interactive control by users through user profile edits as we do.

In this regard, Mysore et al. [37, LACE] who introduce editable user profiles for recommendation tasks presents closely related work to CtrlCE. However, our work differs substantially: we focus on search tasks which bring additional challenges for controllability compared to recommendation setups that lack user queries (Section 3), CtrlCE introduces a calibrated mixing model which highlights queries where control is necessary, and we show how performant cross-encoder models can be controllably personalized. Additionally, our ablation experiments compare to LACE in Section 5.3 and show significantly improved performance by CtrlCE.

Calibrated retrievers. The calibrated mixing model used in CtrlCE ties our work to a small body of work on calibrated ranking models – this work aims to train ranking models that produce confidence scores alongside relevance scores or produces scores that are proportional to ranking model performance, our work relates to the latter. While we build on Yan et al. [54] who train scale-calibrated ranking models for CTR systems, we leverage scale-calibration for facilitating user control in personalized search. Other work has explored probabilistic uncertainty estimation in retrievers through joint training of retrievers in RAG systems [18], Monte Carlo dropout [15], and Gaussian query and document embeddings [58]. In contrast with probabilistic uncertainty estimation, our mixing model produces calibrated scores through regularization and does not require extensive changes to training, model architecture, or additional inference costs. Finally, our mixing model may be seen as a query performance prediction (QPP) model [4] – in this sense, our work represents the first approach using a QPP model for improving personalized search.

3 Problem Formulation

We consider a personalized search problem where a user $u \in \mathcal{U}$, submits a query q to a retrieval system, for retrieving documents from a document collection $\mathcal{D}$ shared with other users. Each user is associated with a user profile $\mathcal{P}_u$ constructed from their historical documents $D_u = \{d_u^i\}_{i=1}^{N_u}$. Historical documents are assumed to capture users' interests and may be documents that users have authored, read, clicked, etc in the past. The retrieval system, f_{Ret}, is tasked with producing a ranking over $\mathcal{D}$ personalized to the user u as $R_u = f_{\text{Ret}}(\mathcal{P}_u, q, \mathcal{D})$. In practice, we take f_{Ret} to be a re-ranking model ranking top K documents from a first-stage retriever.

Given the value of control over personalization [19, 26, 29], we are interested in allowing users control over R_u by manipulating $\mathcal{P}_u$. Such a controllable model should fulfill the following goals:

D1: Communicate interests to the user: The profile should be readable by users to allow edits to it. Specifically, the profile $\mathcal{P}_u$ must communicate the interests of u as represented in D_u. Importantly, we only require our transparent user profile to facilitate interactive control over R_u, without requiring a fully transparent model.

D2: Control retrieval via profile interactions: The profile and model should support edit operations which are reflected in the rankings over documents $R'_u = f(\mathcal{P}'_u, q, \mathcal{D})$. While R_u may trivially be updated via edits to the query, profile interactions can change longer term interests [9], or allow clarifications in complex exploratory searches [42] where changes to the query may not be obvious. Finally, we require f_{Ret} to support retrieval in the absence of any personalization $f_{\text{Ret}}(q, \mathcal{D})$ given that some users might not desire personalization for search or have any historical documents.

D3: Solicit user input when necessary: We require f_{Ret} to identify when profile interactions are likely to be meaningful so that user feedback can be obtained only when necessary. This follows from findings that all queries do not require personalization [2, 48]. Therefore, user edits to $\mathcal{P}_u$ may not always be beneficial.

D4: Performant retrieval: Finally, we require performant retrieval before and after profile interactions since users desire a balance between automated predictions and control [30, 55].

Profile Design. For $\mathcal{P}_u$, we explore two designs – (1) A concept based user profile consisting of a set of natural language concepts, $\mathcal{P}_u = \{k_1, \dots k_P\}$ inferred from D_u, and (2) An item based user profile, $\mathcal{P}_u = \{d_u^1, \dots d_u^P\}$, which directly represents user interests with D_u. Compared to item based profiles, concept based profiles are more succinct and readable – The concepts represent sets of items enabling efficient user interaction and provide short natural language descriptions improving readability. They have also been found to be an intuitive representation for user interaction in prior work [6, 13]. On the other hand, item based representations are finer grained and promise stronger performance. However, both designs are controllable and allow users to express their preferences for controlling personalization – we discuss this in Section 4.3.

4 Proposed Approach

For controllable personalized search, we present CtrlCE, a language model based cross-encoder model personalized using an editable memory constructed from user items. We base CtrlCE on cross-encoders because of their strong performance in search tasks, strong generalization ability across domains [49], and standard use as re-ranking models [32]. To personalize our cross-encoder, we augment it with a *editable memory* constructed from a user's historical documents. To ensure that CtrlCE remains controllable and performant we introduce three key novelties: (1) We introduce an **embedding cross-encoder** which learns separate yet *contextualized* query and document embeddings allowing the CtrlCE cross-encoder to interact with a multi-vector editable user memory. This is in contrast to standard cross-encoders which learn a *fused* query-document representation from `CLS` tokens of pre-trained language models. (2) We construct **editable user profiles** based on dense retriever embeddings. We consider two user profile designs, concept and item-based (see Section 3). While item-based user representations are naturally transparent and editable by users, we introduce *concept-value user representations* for our concept-based user profiles. This user representation pairs each concept in a user profile with a *personalized concept value* computed from user documents. The concept values may be seen as labeled cluster centroids of user documents. (3) Finally, since we are interested in obtaining user edits to $\mathcal{P}_u$ only when necessary, we introduce a **calibrated mixing model** which learns to combine query-document scores from a cross-encoder with user-document scores obtained from user profiles (Figure 1). Besides combining these scores, the mixing models' scores highlight the queries that are likely to benefit from personalization and may in turn benefit from user edits. Next, we describe CtrlCE, the editable memory/user profile, and the training procedure that results in our calibrated mixing model.

4.1 Memory Augmented Cross-encoder

4.1.1 Model Overview. We formulate CtrlCE as a re-ranking model for documents: $s_d = f_{\text{Ret}}(\mathcal{P}_u, q, d)$. We assume access to a user query q, a user profile $\mathcal{P}_u$ constructed from user documents D_u, and the top K candidate documents $d \in \mathcal{D}$ from a first stage ranking model. CtrlCE computes document relevance (s_d) using two scores, query-document relevance from an embedding cross-encoder f_{CE} and user-document relevance from user memory f_{Mem}. These are combined with a calibrated mixing model g_{Mix}:

$$s_d = w \cdot s_d^q + (1-w) \cdot s_d^u = w \cdot f_{\text{CE}}(q, d) + (1-w) \cdot f_{\text{Mem}}(\mathcal{P}_u, q, d) \quad (1)$$

Here, f_{CE} is formulated as an embedding cross-encoder [53] and f_{Mem} computes a score for candidate d based on the interaction between a cross-encoded document embedding and the user profile/memory. Note that in contrast with standard cross-encoders [32], an embedding cross-encoder learns *separate* yet cross-encoded query and document representations, $\mathbf{q}$ and $\mathbf{d}$. This allows f_{CE} to interact with $\mathcal{P}_u$ in f_{Mem}. Therefore, CtrlCE benefits from the strong performance of cross-encoders and yet allows a cross-encoder to interact with user memory for personalization. We formulate f_{Mem} as a multi-vector user representation, i.e. $\mathcal{P}_u$ is represented as a set of P vectors $\mathbf{V}_{i=\{1\dots P\}}$.

Next, the two scores are combined using weight w from a mixing model g_{Mix}. This is trained with a calibrated training objective (Section 4.2) which ensures that w remains proportional to the ranking performance of f_{CE} and serves as a *performance predictor* for it. This allows g_{Mix} to be used by system designers to obtain user edits when CtrlCE relies more on f_{Mem} (D3 of Section 3). Finally, decomposing s_d into two scores supports "no personalization" by dropping the user-document score, s_d^u (D2 of Section 3).

Specifically, f_{CE} inputs concated query and document text into a pre-trained language model (LM) encoder $\texttt{Enc}_{\text{CE}}([q;d])$ and computes the dot product of cross-encoded query and document embeddings: $f_{\text{CE}}(q,d) = \mathbf{q}^T\mathbf{d}$. Both $\mathbf{q}$ and $\mathbf{d}$ are computed as contextualized token embedding averages. Next, user memory embeddings $\mathbf{V}_{i=\{1...P\}}$ are computed from user documents D_u using a pre-trained LM-based memory encoder $\texttt{Enc}_{\text{Mem}}$ in an offline memory construction stage (see Section 4.1.2,4.1.3). Then, f_{Mem} is computed as: $\texttt{max}_{i\in\{1...P\}}\mathbf{d}^T\mathbf{V}_i$. Finally, the mixing weights w, are obtained as a function of the query $\mathbf{q}$, the length of the user profile and query (in tokens), and the model scores s_d^q and s_d^u as: $w = \texttt{sigmoid}(\text{MLP}([\mathbf{q}, \texttt{len}(q), P, s_d^q, s_d^u]))$. This design builds on the intuition that these features are likely to help predict the performance for f_{CE} building on prior work in query performance prediction for non-personalized search [4]. This gives:

$$s_d = w \cdot \mathbf{q}^T\mathbf{d} + (1-w)\cdot \texttt{max}_{i\in\{1...P\}}\mathbf{d}^T\mathbf{V}_i \tag{2}$$

$$w = g_{\text{Mix}}(\mathbf{q}, \texttt{len}(q), P, s_d^q, s_d^u) \tag{3}$$

Next, to ensure that the user profile remains transparent and editable by users (D1, D2) we leverage either a concept-value user representation or an item-based user representation (see Figure 2).

4.1.2 Concept Value Memories. Our concept-value memories represent users with a concept-based user profile $\mathcal{P}_u$ containing P natural language concepts: $\mathcal{P}_u = \{k_1, \dots k_P\}$ – the concepts are succinct descriptors for user documents and are inferred for D_u using dense retrieval of concepts for each user document from a large inventory of interpretable concepts (e.g. Wikipedia categories). Next, the concepts in $\mathcal{P}_u$ are paired with *personalized* concept-values $\mathbf{V}_{i=\{1...P\}}$ which are computed as a function of D_u and $\mathcal{P}_u$. While concepts may be represented simply as embeddings of the concept text, personalized concept values enable stronger personalization performance by using the user documents to compute concept embeddings. Further, since the personalized concept value is also a function of the concept text, any user edits to the concept text also update the personalized concept value. To construct the concept-value memories we use Optimal Transport (OT), a linear programming method for computing assignments between sets of vectors [39], to make a sparse assignment of user documents to concepts. The assigned document content is then used to compute the personalized concept values (Equation (4)). Our work builds on prior work [37] which introduces concept-value memories for controllable recommendations. Here, we show how they can enable control over personalized search models based on powerful cross-encoders.

Specifically, to construct the concept-value memories, we begin by assuming access to a large concept inventory $\mathcal{K}$ (detailed in Section 5.1), and an encoder ($\texttt{Enc}_{\text{Mem}}$) for concepts and user documents D_u that outputs embeddings for them as $\mathbf{K}$ and $\mathbf{S}_D$. Then, $\mathcal{P}_u$ is constructed by retrieving the top-P concepts from $\mathcal{K}$ for the documents in D_u based on their embeddings – a form of zero-shot classification. Notably, $P < N_u$ ensures that the concept-based profile represents a succinct and readable representation of user documents. Next, a sparse and soft assignment of the documents to concepts $\mathbf{Q}_{D\to\mathcal{P}}$ is computed using optimal transport which solves the linear assignment problem: $argmin_{\mathbf{Q}'}\langle dist(\mathbf{S}_D, \mathbf{K}_{\mathcal{P}})\cdot\mathbf{Q}'\rangle$. In the interest of space, we refer readers to prior work for a more detailed presentation of OT [37, 39]. Next, $\mathbf{V}_i$ is computed as an assignment-weighted average of the document content:

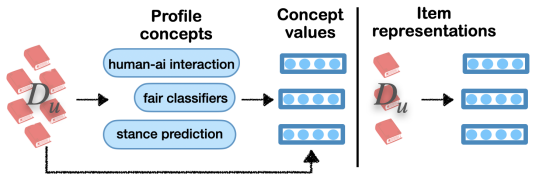


Figure 2: Concept-value memories represent users with concepts and their personalized concept values. Item memories directly represent users with item representations.

$$\mathbf{V}_{i=\{1...P\}} = \frac{1}{\sum_{j=1}^{N_u}\mathbf{Q}_{ji}}\sum_{j=1}^{N_u}\mathbf{Q}_{ji}\cdot\mathbf{S}_j \tag{4}$$

The above design presents some important benefits for controllability, OT computes sparse assignments $\mathbf{Q}$ [39], ensuring that every user document is only assigned to a small number of relevant concepts. Therefore, the concepts partition the documents into soft clusters "tagged" by their concept. This enables users to specify positive or negative preferences for specific concepts, which includes or excludes clusters of documents in generating personalized rankings. Further, user edits to the text of concepts influences their embeddings $\mathbf{K}_{\mathcal{P}}$, which in turn influence $\mathbf{Q}_{D\to\mathcal{P}}$, $\mathbf{V}_{i=\{1...P\}}$, and R_u - allowing edits to reflect in rankings. Finally, OT is readily solved using the Sinkhorn algorithm [16] which runs efficiently on GPUs and can be used inside models trained with gradient descent.

4.1.3 Item Memories. In contrast with concept-value memories, item memories are a simpler user representation where items in D_u are used directly as a user representation, resulting in $\mathbf{V}_{i=\{1...P\}} = \mathbf{S}_D$ with $P = N_u$. Here $\texttt{Enc}_{\text{Mem}}$ directly yields embeddings for D_u. Item memories allow users only to control s_d^u through including or excluding items in $\mathcal{P}_u$. While this can result in cumbersome edits for large profiles, item memories retain finer grained item representations compared to the aggregated representations of concept-value memories which are likely to offer better performance in search tasks. Our experiments (Section 5.2) demonstrate the efficacy of both item and concept-based memories.

4.2 Training

We propose a two-stage procedure for training the embedding cross-encoder ($\texttt{Enc}_{\text{CE}}$) and the calibrated mixing model g_{Mix}. In stage-1 we train our embedding cross-encoder $\texttt{Enc}_{\text{CE}}$ while omitting the mixing model. This results in the relevance scores: $s_d = s_d^q + s_d^u$. Then, in stage-2, we introduce mixing model g_{Mix} to combine s_d^q and s_d^u per Equation (1): $s_d = w\cdot s_d^q + (1-w)\cdot s_d^u$. We use historical user interaction data and a pairwise cross-entropy loss to train $\texttt{Enc}_{\text{CE}}$ and a scale calibrating cross-entropy loss to train g_{Mix}. Our calibrating training procedure ensures that the scores from g_{Mix} don't lie at the extremes of its score range, a known property of MLP-based scoring functions [23, 36, 53], and instead closely tracks the performance of f_{CE}.

Specifically in stage-1, for a user u and query q, we assume access to a relevant document d^+ and a set of M irrelevant documents d^-.

This results in a vector of predicted scores $\mathbf{s}_q = [s_{d^+} \dots s_{d^-}]$ and binary relevance labels $\mathbf{y}_q = [1 \dots 0]$. To train Enc_{CE} we minimize the standard softmax (sm) cross-entropy loss:

$$\mathcal{L}(\mathbf{y}_q, \mathbf{s}_q) = -\sum_{i=1}^{M} \mathbf{y}_q[i] \log \texttt{sm}(\mathbf{s}_q[i]) \tag{5}$$

The parameters of Enc_{CE} are updated while memory representations and Enc_{Mem} are kept fixed throughout training to ensure that model training remains scalable on our GPUs. We initialize Enc_{Mem} with a strong pre-trained LM encoder optimized for dense retrieval and Enc_{CE} with a pre-trained LM encoder.

In stage-2 we train g_{Mix}. Here, we freeze Enc_{Mem} and Enc_{CE} and obtain personalized ranking scores per Equation (1). While we use data identical to that used for training Enc_{CE}, we modify the softmax objective and instead leverage a scale calibrating softmax objective [54]. This modifies the softmax loss by adding an "anchor" example with target score $y_0 \in [0, 1]$, which is a tunable hyperparameter, and logit s_0 set to 0. This results in $\mathbf{s}'_q = [s_{d^+} \dots s_{d^-}, s_0]$ and $\mathbf{y}'_q = [1 - y_0 \dots 0, y_0]$ and the loss:

$$\mathcal{L}(\mathbf{y}'_q, \mathbf{s}'_q) = -\sum_{i=1}^{M} \mathbf{y}'_q[i] \log \frac{e^{\mathbf{s}'_q[i]}}{\sum_j e^{\mathbf{s}'_q[j]} + 1} + y_0 \log \left(\sum_j e^{\mathbf{s}'_q[j]} + 1\right) \tag{6}$$

The insertion of the anchor target y_0 regularizes the scores s_d – penalizing large scores (second term) and preventing smaller scores from being lowered (first term) – ensuring that scores are driven away from the extremes of the score distribution. Further, since only g_{Mix} is trained with the calibrated objective the weights w more smoothly tradeoff the query-document scores s^q_d and the user-document scores s^u_d. As we show in Section 6.2, this results in w being better calibrated with the performance of s^q_d. This calibrated weight w may be used to guide users toward making profile interactions when w indicates that $s^u_d = f_{\text{CE}}(q, d)$ will perform poorly.

4.3 Inference

Retrieval. Performing retrieval with CtrlCE follows a standard two-stage ranking procedure, a first stage ranker retrieves a set of K documents from $\mathcal{D}$, then CtrlCE functions as a re-ranker. It uses s_d (Equation (2)) to re-rank the top K documents and produce a personalized ranked list R_u. To ensure that CtrlCE can be run on standard GPUs, CtrlCE is implemented using 110M parameter transformer LMs, and g_{Mix} is implemented as a 1-layer MLP. Further, because we formulate f_{CE} as an embedding cross-encoder (Section 4.1.1), CtrlCE may be scaled readily to the scale of embedding-based dense retrieval models using recently introduced matrix factorization techniques [53] – we leave this to future work.

Interactive control. Control over personalization in CtrlCE is achieved by interactions with the concept-value or item-based user profile. Given that only some queries may benefit from personalization [2, 48] system designers may only solicit user edits to $\mathcal{P}_u$ for low values of w from the calibrated mixing model g_{Mix}. For example, highly specific lookup queries may not require personalization, on the other hand, exploratory or under-specified queries commonly benefit from personalization [20]. In addition to these interactions, users may also choose to have "no personalization". In CtrlCE, this may be accomplished by re-ranking documents based on query-document relevance (s^q_d) alone.

In interacting with $\mathcal{P}_u$, item profiles offer a more limited range of interactions than concept-value profiles. Item memories support positive/negative selections, whereas concept-value memories support both positive/negative selections and profile edits. 1. Positive/negative selection. Users may choose to include or exclude concept-values (or item embeddings), $\mathbf{V}_{i=\{1\dots P\}}$ ($\mathbf{V}$ in short), to be used for computation of R_u. Positive selection results in the positively selected values, $\mathbf{V}[p, :]$ being used for computing R_u. Similarly, negative selection results in a compliment of the selections $\mathbf{V}[\overline{n}, :]$ being used for computing R_u. Such interactions allow users to include/exclude sets of items or individual items from being used to compute user-document scores s^u_d. 2. Profile edits. Further, for concept-value memories users may also directly change the text of concepts in $\mathcal{P}_u$ triggering re-computation of $\mathbf{V}$ i.e. a reorganization of documents in D_u. These edits may be to edit incorrectly inferred concepts in $\mathcal{P}_u$ or add missing concepts. In experiments, we refer to the item and concept-value profile variants of CtrlCE as $\text{CtrlCE}_{\text{It}}$ and $\text{CtrlCE}_{\text{CV}}$ respectively.

Finally, note that the design of CtrlCE allows efficient updates to R_u based on both "no personalization" and the profile interactions outlined above. This follows from all the score computations in CtrlCE being based on dot-products (Equation (2)). Further, our use of an embedding cross-encoder, and our multi-vector user profiles means that representations can be cached per query and updated rankings produced only using efficient computations.

5 Experiments

We evaluate CtrlCE on four datasets of personalized search constructed from four different scientific domains in a public benchmark for personalized search in Section 5.2. Section 5.3 presents a series of ablations for the components in CtrlCE, and Section 6 we evaluates the controllable components of CtrlCE.

5.1 Experimental Setup

5.1.1 Datasets. We use a public benchmark for personalized search [7], consisting of queries, user documents, and document collections from four scientific domains: computer science (Comp Sci), physics (Physics), political science (Pol Sci), and psychology (Psych). Train and test splits are created temporally such that the test set consists of the most recent queries across the dataset. Each dataset contains 150k-500k training queries, 5k-12k test queries, and profiles between 20-300 documents. We use the title and abstract text to represent documents and report performance using NDCG@10 and note that MRR follows identical trends.

5.1.2 Baselines. We consider a range of standard personalized and non-personalized baselines spanning sparse retrieval, dense retrieval, cross-encoders, and ensemble methods. Our non-personalized approaches span: sparse retrieval with BM25, weakly supervised dense retrieval with Contriver, supervised dense retriever trained on 1 billion pairs (MPNet-1B, HF: all-mpnet-base-v2), and supervised dense retrieval trained on 250M community question answer sites (MPNet-CQA, HF: multi-qa-mpnet-base-cos-v1) – this is noted to be valuable training data for dense retrievers [36]. Finally, CrossEnc is a standard cross-encoder trained on the same

Table 1: CtrlCE is compared against non-personalized (first block) and personalized (second block) approaches. Bold indicates CtrlCE improvement over CrossEnc and * indicates statistical significance with a two-sided t-test at $p < 0.05$.

	Comp Sci	Physics	Pol Sci	Psych
Model	NDCG@10	NDCG@10	NDCG@10	NDCG@10
BM25	0.2245	0.2688	0.2407	0.2393
Contriver	0.1658	0.1795	0.1932	0.1887
MPNet-1B	0.2168	0.1885	0.2306	0.2407
MPNet-CQA	0.1969	0.2093	0.2153	0.2187
CrossEnc	0.2934	0.3330	0.3102	0.3346
rf(BM25,QA)	0.2849	0.3272	0.2894	0.3112
rf(BM25,MPNet-1B,QA)	0.3115	0.3481	0.3141	0.3485
RetAugCE	*0.3244*	*0.3691*	*0.3384*	*0.3778*
$\text{CtrlCE}_{\text{It}}$	**0.3223***	**0.3657***	**0.3378***	**0.3704***
$\text{CtrlCE}_{\text{CV}}$	**0.3118***	**0.3583***	**0.3310***	**0.3614***

data as CtrlCE while ignoring the user documents D_u. CrossEnc is initialized with MPNet-base [45], and the query-document score is produced by passing the CLS representation through an MLP. CrossEnc is the closest comparator to CtrlCE since it is a standard non-personalized cross-encoder. We include the details of baselines in our code repository.

Our personalized approaches use prior personalized dense retrieval models and personalized cross-encoder models: rf(BM25, QA): A rank-fusion approach that learns a weighted combination of BM25 scores and scores from a Query Attention (QA) based user modeling approach. Importantly, QA is a key component of effective personalization in prior work on personalized search, [22, HRNN-QA], [2, ZAM], and [27, EDAM]. QA scores candidate documents based on their dot product similarity to the weighted average of user documents. The weights for user documents are computed as dot-product attentions between query and document representations from MPNet-1B. rf(BM25, QA, MPNet-1B): This adds dense retrieval scores from MPNet-1B to rf(BM25, QA). RetAugCE: A cross-encoder personalized with retrieval-augmentation [44]. It inputs, query, candidate, and the top-1 document most similar to the query from D_u. This approach follows the state-of-the-art personalized cross-encoder model for product search [17, CoPPS], however given the longer length of documents in our datasets compared to e-commerce products, we use retrieval to reduce input sequence lengths. We use the MPNet-CQA model for retrieving the top-1 document. Finally, note that we primarily aim to demonstrate that CtrlCE results in strong performance compared to several reasonable and strong baselines while remaining controllable (Section 6). We leave the exploration of strategies such as contrastive self-supervised training [17, 62] for establishing SOTA performance in controllable cross-encoder models to future work.

5.1.3 Implementation Details. In CtrlCE we initialize Enc_{CE} with MPNet-base [45] and Enc_{Mem} with MPNet-CQA for both $\text{CtrlCE}_{\text{It}}$ and $\text{CtrlCE}_{\text{CV}}$. We formulate g_{Mix} as an MLP with one hidden layer of 386 dimensions and use a tanh non-linearity. For first-stage ranking, we use BM25 and re-rank $K = 200$ documents per query. Further, in $\text{CtrlCE}_{\text{CV}}$ for constructing concept-based user profiles for a concept inventory $\mathcal{K}$ we use a collection of computer science concepts from [34] and Wikipedia categories for Physics, Pol Sci, and Psych. Our code repository includes additional details.

5.2 Experimental Results

Baseline performance. We begin by examining baseline performance in Table 1. We see that personalized models that ensemble various sparse and dense personalized and non-personalized models with rank fusion ($\texttt{rf}(\cdot)$) outperform non-personalized sparse and dense models (rows BM25 to MPNet-CQA). However, a non-personalized cross-encoder (CrossEnc) outperforms all other non-personalized models and approaches the performance of personalized models based on Query Attention (QA). The strong performance of *non-personalized* cross-encoders in personalized search has also been noted in prior work [35]. Next, we note that a cross-encoder personalized with retrieval augmentation, RetAugCE outperforms the non-personalized CrossEnc while incurring greater inference costs.

CtrlCE performance. For the proposed CtrlCE models we first note that both CtrlCE variants outperform the non-personalized CrossEnc with improvements of 6.4-10.6% across evaluation metrics and datasets. This indicates the proposed memory-augmented cross-encoder to be effective at personalization. Next, we compare CtrlCE models to personalized methods based on Query Attention (QA). Here, we note CtrlCE models to outperform personalized ensemble methods. Finally, while CtrlCE performs at par with RetAugCE (no statistically significant difference with $\text{CtrlCE}_{\text{It}}$), it does not outperform it. However, as we show in Section 6, RetAugCE does not allow control over personalization – lacking the ability to identify when control interactions are necessary and being unable to support the "no personalization" action (Sections 6.2 and 6.1). Therefore, CtrlCE performs at par with state-of-the-art approaches while remaining controllable.

$\text{CtrlCE}_{\text{It}}$ vs $\text{CtrlCE}_{\text{CV}}$. Here, we examine the difference between item and concept-value memories, $\text{CtrlCE}_{\text{CV}}$ and $\text{CtrlCE}_{\text{It}}$. We see $\text{CtrlCE}_{\text{It}}$ to consistently outperform $\text{CtrlCE}_{\text{It}}$ by a small margin. We hypothesize that this is due to the nature of the search task where most queries seek specific items rather than being exploratory. As a consequence, the finer-grained item representations that $\text{CtrlCE}_{\text{It}}$ retains allow higher precision in retrieval at the expense of more tedious interactions for controllable personalization. However, recall from Sections 3 and 4.3 that the concept-based profiles of $\text{CtrlCE}_{\text{CV}}$ provide a richer set of profile interactions and a more compact and readable user profile – in applications where this is important practitioners may choose to use $\text{CtrlCE}_{\text{CV}}$ over $\text{CtrlCE}_{\text{It}}$. Next, we illustrate the performance resulting from the various components of CtrlCE in a series of ablations.

5.3 Ablation Study

Table 2 presents an ablation indicating the performance of the various model components of CtrlCE. We present results for both $\text{CtrlCE}_{\text{It}}$ and $\text{CtrlCE}_{\text{CV}}$. Further, we only present results with NDCG@10 in the interest of space, noting that MRR follows the same trends. We report statistical significance compared against $\text{CtrlCE}_{\text{It}}$/$\text{CtrlCE}_{\text{CV}}$ with * using a two-sided t-test at $p < 0.05$.

No user-document score. We begin by examining a test time only change – after training CtrlCE as described in Section 4.2, we

Table 2: CtrlCE components ablated for item ($\text{CtrlCE}_{\text{It}}$) and concept-value ($\text{CtrlCE}_{\text{CV}}$) memories.

	Comp Sci	Physics	Pol Sci	Psych
Model	NDCG@10	NDCG@10	NDCG@10	NDCG@10
$\text{CtrlCE}_{\text{It}}$	0.3223	0.3657	0.3378	0.3704
– no f_{Mem}	0.3010*	0.3440*	0.3164*	0.3465*
– no f_{CE}	0.1994*	0.2424*	0.1997*	0.2450*
– no g_{Mix}	0.3078*	0.3452*	0.3236*	0.3517*
– no calibration	0.3065*	0.3633	0.3277*	0.3623*
$\text{CtrlCE}_{\text{CV}}$	0.3118	0.3583	0.3310	0.3614
– no f_{Mem}	0.2936*	0.3344*	0.3151*	0.3368*
– no f_{CE}	0.1675*	0.1981*	0.1792*	0.2060*
– no g_{Mix}	0.3061	0.3390*	0.3278	0.3399*
– no calibration	0.3121	0.3602	0.3326	0.3625

produce personalized rankings R_u only using the query-document scores produced by f_{CE} per Equation (1) (– no f_{Mem}, Table 2). We see that both $\text{CtrlCE}_{\text{It}}$ and $\text{CtrlCE}_{\text{CV}}$ see consistent drops in performance indicating the value provided by personalization with editable user memory.

No query-document score. Having trained CtrlCE, we examine the personalized rankings produced using only the user-document scores produced by f_{Mem} (– no f_{CE}, Table 2). This ablation mirrors the proposed approach of Mysore et al. [37, LACE] for controllable recommendations. As expected, we see that the lack of query-document scores results in a large drop in performance indicating the value of CtrlCE over f_{Mem} alone.

No mixing model. In this experiment, we train a memory augmented cross-encoder without the mixing model g_{Mix} (– no g_{Mix}, Table 2) producing test time rankings using a simple summation of query-document and user-document scores: $s_d = s^q_d + s^u_d$. This may also be seen as a model resulting from stage-1 training alone. Here, we see that this approach consistently underperforms $\text{CtrlCE}_{\text{It}}$ and $\text{CtrlCE}_{\text{CV}}$, indicating the value of g_{Mix} for ranking performance.

No calibrated objective. Finally, we consider a model similar to CtrlCE, trained with g_{Mix} with two-stage training but lacking in the calibrated softmax objective of Section 4.2 and instead using a standard softmax objective for both training stages (– no calibration, Table 2). We see that omission of the calibrated objective results in a similar performance to CtrlCE showing calibrated training to not harm performance. In Section 6.2 we show how the calibrated training results in a stronger correlation between w and the performance of f_{CE} indicating its value for controllable personalization.

6 Interaction Evaluation

CtrlCE supports control over personalized search in three ways: (1) support for a "no personalization" setting, (2) effectively highlighting queries where user edits to $\mathcal{P}_u$ may be necessary, (3) control over item or concept based profiles $\mathcal{P}_u$. Here, we demonstrate how CtrlCE effectively supports these control actions. In Section 6.1, we evaluate CtrlCE's ability to support a "no personalization" action compared against the RetAugCE model. In Section 6.2 we demonstrate that our mixing model closely tracks the performance of f_{CE} allowing CtrlCE to obtain user edits only when necessary.

Table 3: CtrlCE compared to RetAugCE for the control action of "no personalization". MRR follows identical trends.

	Comp Sci	Physics	Pol Sci	Psych
Model	NDCG@10	NDCG@10	NDCG@10	NDCG@10
CrossEnc	0.2934	0.3330	0.3102	0.3346
RetAugCE	0.3244	0.3691	0.3384	0.3778
– no personalization	0.2264	0.2699	0.2039	0.2949
$\text{CtrlCE}_{\text{It}}$	0.3223	0.3657	0.3378	0.3704
– no personalization	0.3010	0.3440	0.3164	0.3465
$\text{CtrlCE}_{\text{CV}}$	0.3118	0.3583	0.3310	0.3614
– no personalization	0.2936	0.3344	0.3151	0.3368

Finally, in Section 6.3 we demonstrate the controllability provided by editable item and concept-based profiles in a user study.

6.1 "No personalization" evaluation

6.1.1 Setup. CtrlCE accomplishes "no personalization" by ranking documents using query-document score s^q_d and dropping user-document score s^u_d. We compare this to the retrieval-augmented baseline RetAugCE. Here, "no personalization" is accomplished by only inputting query-document pairs into RetAugCE, dropping a retrieved document from D_u. In this setup, a controllable model must perform at par with a non-personalized cross-encoder.

6.1.2 Results. In Table 3 we see that dropping personalization from CtrlCE reverts it to perform similar to a non-personalized cross-encoder CrossEnc, indicating its ability to maintain performance while accomplishing a "no personalization" control action. On the other hand, RetAugCE sees a large drop in performance from not using the retrieved context indicating it to be a much harder model to control. Note also, that ranking for "no personalization" may be accomplished efficiently through per-query cached representations, not requiring repeated forward passes through CtrlCE.

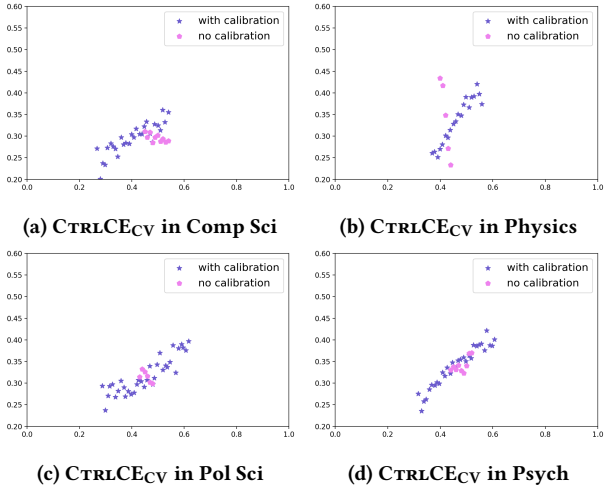
6.2 Selective control evaluation

To demonstrate that CtrlCE selectively highlights the queries which would benefit from user control we show how the mixing model (g_{Mix}) also serves as a performance predictor for the cross-encoder model (f_{CE}) by evaluating its calibration performance – i.e the ability of g_{Mix} scores to be proportional to the ranking performance of f_{CE}. This enables system designers to use g_{Mix} to identify queries where CtrlCE will rely more heavily on f_{Mem} and where user over $\mathcal{P}_u$ may improve performance. We report results in Table 4 and Figure 3. Finally, we present a small-scale case study to highlight the queries which g_{Mix} selects for personalization in Table 5.

6.2.1 Setup. We measure calibration performance with the Pearson correlation between the score w produced by g_{Mix} for the top-1 document for each query and the NDCG@10 metric for f_{CE}. To compute this metric, we bucket all the queries in our test set into 100 equal-sized buckets based on the value of w for the top retrieved document for a query. Then we compute the average NDCG@10 for the queries within each bucket. Finally, we measure the Pearson correlation between the lower edge of each bucket and the average

Table 4: The Pearson correlation between scores produced by the mixing model g_{Mix} and NDCG@10 for f_{CE}. CtrlCE$_{\text{It}}$ and CtrlCE$_{\text{CV}}$ are compared against the respective models trained without a calibrating objective.

	Comp Sci	Physics	Pol Sci	Psych
Model	f_{CE}	f_{CE}	f_{CE}	f_{CE}
CtrlCE$_{\text{It}}$	**0.81**	**0.90**	**0.73**	**0.86**
– no calibration	0.13	0.54	0.48	0.62
CtrlCE$_{\text{CV}}$	**0.73**	**0.90**	**0.76**	**0.95**
– no calibration	0.22	0.92	0.42	0.38



(a) CtrlCE$_{\text{CV}}$ in Comp Sci **(b) CtrlCE$_{\text{CV}}$ in Physics**

(c) CtrlCE$_{\text{CV}}$ in Pol Sci **(d) CtrlCE$_{\text{CV}}$ in Psych**

Figure 3: Scores produced by the mixing model g_{Mix} used to combine f_{CE} and f_{Mem} (Equation (1)) plotted against the NDCG@10 for f_{CE}. CtrlCE$_{\text{CV}}$ (blue) is compared against a model trained without a calibrated objective (pink). Our calibrated objective ensures that g_{Mix} scores are proportional to f_{CE} performance. CtrlCE$_{\text{It}}$ shows identical trends.

NDCG@10 metric within each bucket while excluding buckets with fewer than 50 queries. We also plot these values in Figure 3. We compare CtrlCE (blue) to models trained without the calibrated objective of Section 4.2 (pink).

6.2.2 Results. In Table 4 we note that the calibrated training of CtrlCE results in g_{Mix} being consistently linearly correlated with the performance of f_{CE}. We also note from Figure 3 that in the absence of calibrated training the g_{Mix} scores (pink) rarely track the performance of f_{CE}. This not only results in poor performance (see Table 2, "– no calibration" row) but also indicates that the scores would not be useful for selectively soliciting user edits. The strong correlation of the g_{Mix} with f_{CE} indicates its potential for guiding users to provide control interactions for $\mathcal{P}_u$ only when f_{CE} alone is likely to underperform.

6.2.3 Case study. In Table 5 we include a small-scale case study of g_{Mix} outputs to illustrate queries selected for personalization. Here we manually examine the queries that receive among the highest and lowest weights from g_{Mix}. For these queries, we examined the ranked documents by f_{CE} and CtrlCE (i.e $w \cdot f_{\text{CE}} + (1-w) \cdot f_{\text{Mem}}$) and their relevance judgments. Queries with a low g_{Mix} predicted weight are likely to require personalization and in turn benefit from user edits to $\mathcal{P}_u$. We show examples from CtrlCE$_{\text{It}}$ predictions for computer science and political science domains given the relative ease of understanding these domains.

Table 5: Example queries which g_{Mix} predicts as likely to require personalization (red). These queries have improved performance in CtrlCE over f_{CE} ("Gain"). On the other hand, queries predicted as likely to perform well with f_{CE} alone see no improvement from personalization (green).

ID	Query text	g_{Mix} score	Gain NDCG@10	Query type
CS1	"normal integration survey"	0.15	+0.41	exploratory
CS2	"tense data mining for data fusion"	0.08	+0.40	ambiguous
CS3	"scale aware cnn pedestrian detection"	0.64	+0.00	unambiguous
CS4	"katyusha acceleration sgd"	0.63	-0.01	unambiguous
PS1	"worldwide research on probiotics"	0.06	+0.25	exploratory
PS2	"consumers' willingness to pay for hale"	0.12	+0.22	ambiguous
PS3	"patent pools, competition, and innovation, evidence from us industries"	0.50	+0.00	unambiguous
PS4	"co2 emissions in chinas lime industry"	0.45	-0.01	unambiguous

In Table 5 we see that g_{Mix} commonly scores exploratory queries (CS1 , PS1) and ambiguous look-up queries (CS2 , PS2) with a low score. In both these cases personalization based on f_{Mem} plays a greater role in producing ranked documents. In both these cases we see that f_{Mem} helps CtrlCE outperform f_{CE} alone (positive "CCE gain"). We also see that unambiguous look-up queries that seek more specific relevant documents (CS3-4 , PS3-4) result in higher scores from g_{Mix} for f_{CE}. Consequently f_{CE} and CtrlCE perform nearly identically in these queries. This illustrates that g_{Mix} successfully identifies queries that benefit from personalization and could benefit from user edits to $\mathcal{P}_u$. Finally, we note that not all queries with low values of w may benefit from edits to $\mathcal{P}_u$, e.g. some could benefit from query reformulation. We leave the exploration of finer-grained performance prediction to future work. Nevertheless, our experiments show that g_{Mix} remains calibrated and enables obtaining user control for $\mathcal{P}_u$ only when necessary.

6.3 Editability evaluation

We evaluate the controllability of the editable memories of CtrlCE in a user study with realistic queries and user profiles. We ran our user study with expert annotators interacting with interactive prototypes of CtrlCE$_{\text{It}}$ and CtrlCE$_{\text{CV}}$. Through our study, we aim to answer the research question: Are users able to improve the search performance for CtrlCE$_{\text{It}}$ and CtrlCE$_{\text{CV}}$ through interaction with user profiles, $\mathcal{P}_u$? Table 6 and Section 6.3.3 present our results.

6.3.1 Data and Model Setup. Our user study was run as an expert-driven user study with two computer science annotators. Our annotators interacted with an interactive prototype of CtrlCE and judged 45 realistic queries and their associated user profiles selected from our Comp Sci evaluation dataset. Because we are primarily interested in evaluating the controllability of editable memories in CtrlCE we selected the 200 queries where g_{Mix} scores indicated that the query was most likely to benefit from personalization. This

Table 6: User study evaluation of CtrlCE$_{It}$ and CtrlCE$_{CV}$ demonstrating their ability to enable users to improve search performance through interactions with a user profile.

	CtrlCE$_{It}$		CtrlCE$_{CV}$	
	NDCG@20	R@20	NDCG@20	R@20
Initial	0.7169	0.7549	0.7169	0.7549
Tuned	**0.7537**	**0.8247***	**0.7303**	**0.8139****
Gain	+0.0368	+0.0698	+0.0134	+0.0590

set of queries is also likely to benefit from user control to $\mathcal{P}_u$. Next, to prevent burdensome annotations we only retained queries that had between 20-50 historical documents (D_u). Next, for the 200 queries, our expert annotators indicated which queries were topically familiar and of research interest to them – allowing them to reasonably stand in as the original users for judging CtrlCE rankings. This resulted in 45 queries for our user study. Our expert annotators were computer science researchers and part of the authorship team. Both annotators had interests in design and AI, had experience reading research papers, and were compensated for their time. We opted for an expert-driven user study instead of one with real users due to limits on the length of our user study. Because not all queries require personalization a study that relies on real users would need to be run for a longer period to ensure that users organically made sufficient queries which required personalization and edits to their user profile. Instead, we conduct careful evaluations of all parts of CtrlCE (Section 6.1-6.3.2) and leave end-to-end evaluations over longer times to future work.

6.3.2 Study Procedure. To evaluate the controllability of the CtrlCE models our user study was conducted in three phases: (A) Our annotators rated an initial ranked list from CtrlCE$_{It}$ and CtrlCE$_{CV}$ for the relevance of documents, (B) Next, they tuned user profiles with interactions (Section 4.3) which they judged would improve the initial ranked list, and (C) They rated an updated ranked list of documents produced as a result of the control action. The ratings gathered from this procedure were used to measure the effect of CtrlCE's editable memories on search performance. To ensure that our annotators made reasonable and unbiased ratings, we included a guideline generation and agreement measurement phase in our study. Further, we release the guidelines and generated ratings in our code release to ensure transparency in the process. To measure agreement, both annotators rated 15 shared queries and the ranked lists from CtrlCE$_{It}$. Then, they met and resolved their rating disagreements, and created an adjudicated set of ratings and rating guidelines. We noted annotator agreement with the adjudicated ratings to be Cohens $\kappa = 0.81$ and a rank-correlation of $\rho = 0.84$, indicating the annotation guideline to be sound and the annotators in agreement. The guideline was then applied to 30 queries not used for agreement measurements, and Table 6 reports these. In making their ratings annotators first familiarized themselves with the user profile and query, and then rated $K = 30$ ranked documents on a 3-point scale. They reported spending 15-20 minutes per query.

6.3.3 Results. Table 6 presents the primary result of our user study. We report NDCG and Recall at deeper ranks, $K = 20$ to illustrate how control over personalization improves the users' ability to explore collections – this is commonly done through exploring to deeper rank positions. Because we gather ratings for $K = 30$ we report Recall@20. We report statistical significance at $p < 0.05$ and $p < 0.10$ with a paired t-test, denoted as * and **.

From Table 6 we note that users were able to improve the performance of CtrlCE$_{It}$ by 5-10% and CtrlCE$_{CV}$ by 2-8% across metrics, with statistically significant improvements for R@20. Based on this we answer our research question in the affirmative - users are able to effectively interact with editable memories in CtrlCE to improve search performance. Further, the larger improvements in R@20 indicate that control interactions improve exploratory ability than precision-oriented performance. We also note that larger-scale studies may be needed to establish statistically significant improvements in NDCG@20. Finally, our user study did not probe aspects of interface usability or user trust from controllable personalization, future work may probe these aspects further in longer-term deployments in realistic application contexts.

7 Conclusion

In this paper, we introduce CtrlCE, a memory-augmented cross-encoder for controllable personalized search. To facilitate control while achieving strong ranking performance, we augment expressive cross-encoder models with editable memories of user documents. To ensure that our expressive cross-encoder is able to interact with a multi-vector user memory we formulate it as a novel embedding cross-encoder. Further, we introduce a calibrated mixing model that indicates when queries benefit from personalization and in turn from greater user interactions. In experiments on four scientific domains, we demonstrate CtrlCE to improve upon a wide variety of standard prior methods spanning, sparse, dense, cross-encoder, and personalized approaches. We demonstrate the controllability of CtrlCE through experiments demonstrating its ability to support a "no personalization" interaction. In calibration evaluations and a case study, we demonstrate its ability to seek user interaction only when necessary. Finally, in a user study we demonstrate that when user interactions are sought, interaction with item and concept-based profiles successfully improves performance.

Acknowledgments

We thank anonymous reviewers for their feedback. This work was partly supported by the Center for Intelligent Information Retrieval, IBM Research AI through the AI Horizons Network, CZI under the project Scientific Knowledge Base Construction, NSF grant number IIS-2106391, Amazon Alexa Prize Competition, Lowes, Adobe, Google, and Microsoft. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsors.

References

[1] Jae-wook Ahn, Peter Brusilovsky, and Shuguang Han. 2015. Personalized Search: Reconsidering the Value of Open User Models. In *IUI*. 11 pages. https://doi.org/10.1145/2678025.2701410

[2] Qingyao Ai, Daniel N. Hill, S. V. N. Vishwanathan, and W. Bruce Croft. 2019. A Zero Attention Model for Personalized Product Search. In *CIKM*. 10 pages. https://doi.org/10.1145/3357384.3357980

[3] Qingyao Ai, Yongfeng Zhang, Keping Bi, Xu Chen, and W. Bruce Croft. 2017. Learning a Hierarchical Embedding Model for Personalized Product Search. In

SIGIR. 10 pages. https://doi.org/10.1145/3077136.3080813
[4] Negar Arabzadeh, Chuan Meng, Mohammad Aliannejadi, and Ebrahim Bagheri. 2024. Query Performance Prediction: From Fundamentals to Advanced Techniques. In *Advances in Information Retrieval*, Nazli Goharian, Nicola Tonellotto, Yulan He, Aldo Lipani, Graham McDonald, Craig Macdonald, and Iadh Ounis (Eds.). Springer Nature Switzerland, Cham, 381–388.
[5] Grigor Aslanyan, Aritra Mandal, Prathyusha Senthil Kumar, Amit Jaiswal, and Manojkumar Rangasamy Kannadasan. 2020. Personalized Ranking in ECommerce Search. In *The Web Conference*. 2 pages. https://doi.org/10.1145/3366424.3382715
[6] Krisztian Balog, Filip Radlinski, and Shushan Arakelyan. 2019. Transparent, scrutable and explainable user models for personalized recommendation. In *SIGIR*. 265–274.
[7] Elias Bassani, Pranav Kasela, Alessandro Raganato, and Gabriella Pasi. 2022. A Multi-Domain Benchmark for Personalized Search Evaluation. In *CIKM*. 6 pages. https://doi.org/10.1145/3511808.3557536
[8] Michael Bendersky, Xuanhui Wang, Donald Metzler, and Marc Najork. 2017. Learning from User Interactions in Personal Search via Attribute Parameterization. In *WSDM*. 9 pages. https://doi.org/10.1145/3018661.3018712
[9] Paul N. Bennett, Ryen W. White, Wei Chu, Susan T. Dumais, Peter Bailey, Fedor Borisyuk, and Xiaoyuan Cui. 2012. Modeling the Impact of Short- and Long-Term Behavior on Search Personalization. In *SIGIR*. 10 pages. https://doi.org/10.1145/2348283.2348312
[10] Keping Bi, Qingyao Ai, and W. Bruce Croft. 2020. A Transformer-Based Embedding Model for Personalized Product Search. In *SIGIR*. 4 pages. https://doi.org/10.1145/3397271.3401192
[11] Keping Bi, Pavel Metrikov, Chunyuan Li, and Byungki Byun. 2021. Leveraging User Behavior History for Personalized Email Search. In *The Web Conference*. 11 pages. https://doi.org/10.1145/3442381.3450110
[12] Fei Cai, Shangsong Liang, and Maarten de Rijke. 2014. Personalized Document Re-Ranking Based on Bayesian Probabilistic Matrix Factorization. In *SIGIR*. 4 pages. https://doi.org/10.1145/2600428.2609453
[13] Shuo Chang, F. Maxwell Harper, and Loren Terveen. 2015. Using Groups of Items for Preference Elicitation in Recommender Systems. In *CSCW*. 12 pages. https://doi.org/10.1145/2675133.2675210
[14] Jennifer Chien and David Danks. 2023. Fairness Vs. Personalization: Towards Equity in Epistemic Utility. *FAccTRec 2023 at RecSys* (2023).
[15] Daniel Cohen, Bhaskar Mitra, Oleg Lesota, Navid Rekabsaz, and Carsten Eickhoff. 2021. Not All Relevance Scores Are Equal: Efficient Uncertainty and Calibration Modeling for Deep Retrieval Models. In *SIGIR*. 11 pages. https://doi.org/10.1145/3404835.3462951
[16] Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *NeurIPS* 26 (2013), 2292–2300.
[17] Shitong Dai, Jiongnan Liu, Zhicheng Dou, Haonan Wang, Lin Liu, Bo Long, and Ji-Rong Wen. 2023. Contrastive Learning for User Sequence Representation in Personalized Product Search. In *KDD*. 10 pages. https://doi.org/10.1145/3580305.3599287
[18] Shehzaad Dhuliawala, Leonard Adolphs, Rajarshi Das, and Mrinmaya Sachan. 2022. Calibration of Machine Reading Systems at Scale. In *ACL Findings*. https://aclanthology.org/2022.findings-acl.133
[19] Cecilia di Sciascio, Eduardo Veas, Jordan Barria-Pineda, and Colleen Culley. 2020. Understanding the Effects of Control and Transparency in Searching as Learning. In *IUI*. 12 pages. https://doi.org/10.1145/3377325.3377524
[20] Susan Dumais, Edward Cutrell, JJ Cadiz, Gavin Jancke, Raman Sarin, and Daniel C. Robbins. 2003. Stuff I've seen: a system for personal information retrieval and re-use. In *SIGIR*. https://doi.org/10.1145/860435.860451
[21] Malin Eiband, Sarah Theres Völkel, Daniel Buschek, Sophia Cook, and Heinrich Hussmann. 2019. When People and Algorithms Meet: User-Reported Problems in Intelligent Everyday Applications. In *IUI*. 11 pages. https://doi.org/10.1145/3301275.3302262
[22] Songwei Ge, Zhicheng Dou, Zhengbao Jiang, Jian-Yun Nie, and Ji-Rong Wen. 2018. Personalizing Search Results Using Hierarchical RNN with Query-aware Attention. In *CIKM*. 10 pages. https://doi.org/10.1145/3269206.3271728
[23] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *ICML* (Sydney, NSW, Australia). 1321–1330.
[24] Viet Ha-Thuc and Shakti Sinha. 2016. Learning to Rank Personalized Search Results in Professional Networks. In *SIGIR*. 2 pages. https://doi.org/10.1145/2911451.2927018
[25] Bryan Horling and Matthew Kulick. 2009. Personalized Search for everyone. https://googleblog.blogspot.com/2009/12/personalized-search-for-everyone.html. Accessed: 12 Jan 2024.
[26] Dietmar Jannach, Sidra Naveed, and Michael Jugovac. 2016. User control in recommender systems: Overview and interaction challenges. In *International Conference on Electronic Commerce and Web Technologies*. Springer, 21–33.
[27] Jyun-Yu Jiang, Tao Wu, Georgios Roumpos, Heng-Tze Cheng, Xinyang Yi, Ed Chi, Harish Ganapathy, Nitin Jindal, Pei Cao, and Wei Wang. 2020. End-to-End Deep Attentive Personalized Item Retrieval for Online Content-sharing Platforms. In *The Web Conference*. https://doi.org/10.1145/3366423.3380051
[28] Lu Jiang, Yannis Kalantidis, Liangliang Cao, Sachin Farfade, Jiliang Tang, and Alexander G. Hauptmann. 2017. Delving Deep into Personal Photo and Video Search. In *CIKM*. 10 pages. https://doi.org/10.1145/3018661.3018736
[29] Yucheng Jin, Bruno Cardoso, and Katrien Verbert. 2017. How do different levels of user control affect cognitive load and acceptance of recommendations?. In *Workshop on Interfaces and Human Decision Making for Recommender Systems at RecSys*, Vol. 1884. CEUR Workshop Proceedings, 35–42.
[30] Bart P. Knijnenburg, Niels J.M. Reijmer, and Martijn C. Willemsen. 2011. Each to His Own: How Different Users Call for Different Interaction Methods in Recommender Systems. In *RecSys*. 8 pages. https://doi.org/10.1145/2043932.2043960
[31] Joseph Konstan and Loren Terveen. 2021. Human-centered recommender systems: Origins, advances, challenges, and opportunities. *AI Magazine* 42, 3 (2021), 31–42.
[32] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2021. Pretrained transformers for text ranking: Bert and beyond. *Synthesis Lectures on Human Language Technologies* 14, 4 (2021), 1–325. https://arxiv.org/pdf/2010.06467.pdf
[33] Jiongnan Liu, Zhicheng Dou, Jian-Yun Nie, and Ji-Rong Wen. 2024. Integrated Personalized and Diversified Search Based on Search Logs. *IEEE Transactions on Knowledge and Data Engineering* 36, 2 (2024), 694–707.
[34] Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. S2ORC: The Semantic Scholar Open Research Corpus. In *ACL*. https://www.aclweb.org/anthology/2020.acl-main.447
[35] Sean MacAvaney, Craig Macdonald, and Iadh Ounis. 2022. Reproducing Personalised Session Search Over the AOL Query Log. In *Advances in Information Retrieval*, Matthias Hagen, Suzan Verberne, Craig Macdonald, Christin Seifert, Krisztian Balog, Kjetil Nørvåg, and Vinay Setty (Eds.). Springer International Publishing, Cham, 627–640.
[36] Aditya Menon, Sadeep Jayasumana, Ankit Singh Rawat, Seungyeon Kim, Sashank Reddi, and Sanjiv Kumar. 2022. In defense of dual-encoders for neural ranking. In *ICML*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (Eds.). https://proceedings.mlr.press/v162/menon22a.html
[37] Sheshera Mysore, Mahmood Jasim, Andrew Mccallum, and Hamed Zamani. 2023. Editable User Profiles for Controllable Text Recommendations. In *SIGIR*. 11 pages. https://doi.org/10.1145/3539618.3591677
[38] Vito Ostuni, Christoph Kofler, Manjesh Nilange, Sudarshan Lamkhede, and Dan Zylberglejd. 2023. Search Personalization at Netflix. In *The Web Conference*. 3 pages. https://doi.org/10.1145/3543873.3587675
[39] Gabriel Peyré, Marco Cuturi, et al. 2019. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning* 11, 5-6 (2019), 355–607. https://www.nowpublishers.com/article/Details/MAL-073
[40] Zhen Qin, Zhongliang Li, Michael Bendersky, and Donald Metzler. 2020. Matching Cross Network for Learning to Rank in Personal Search. In *The Web Conference*. 7 pages. https://doi.org/10.1145/3366423.3380046
[41] Filip Radlinski and Susan Dumais. 2006. Improving Personalized Web Search Using Result Diversification. In *SIGIR*. 2 pages. https://doi.org/10.1145/1148170.1148320
[42] Tuukka Ruotsalo, Jaakko Peltonen, Manuel J. A. Eugster, Dorota Głowacka, Patrik Floréen, Petri Myllymäki, Giulio Jacucci, and Samuel Kaski. 2018. Interactive Intent Modeling for Exploratory Search. *ACM Trans. Inf. Syst.* 36, 4, Article 44 (oct 2018), 46 pages. https://doi.org/10.1145/3231593
[43] Sara Salehi, Jia Tina Du, and Helen Ashman. 2015. Examining Personalization in Academic Web Search. In *HT*. 9 pages. https://doi.org/10.1145/2700171.2791039
[44] Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2023. LaMP: When Large Language Models Meet Personalization. arXiv:2304.11406
[45] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. MPNet: Masked and Permuted Pre-training for Language Understanding. In *NeurIPS*. ACM. https://www.microsoft.com/en-us/research/publication/mpnet-masked-and-permuted-pre-training-for-language-understanding/
[46] David Sontag, Kevyn Collins-Thompson, Paul N. Bennett, Ryen W. White, Susan Dumais, and Bodo Billerbeck. 2012. Probabilistic Models for Personalizing Web Search. In *WSDM*. 10 pages. https://doi.org/10.1145/2124295.2124348
[47] Jaime Teevan, Susan T. Dumais, and Eric Horvitz. 2005. Personalizing Search via Automated Analysis of Interests and Activities. In *SIGIR*. 8 pages. https://doi.org/10.1145/1076034.1076111
[48] Jaime Teevan, Susan T. Dumais, and Daniel J. Liebling. 2008. To Personalize or Not to Personalize: Modeling Queries with Variation in User Intent. In *SIGIR*. 8 pages. https://doi.org/10.1145/1390334.1390364
[49] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. In *NeurIPS Datasets and Benchmarks Track (Round 2)*. https://openreview.net/forum?id=wCu6T5xFjeJ
[50] David Vallet and Pablo Castells. 2012. Personalized Diversification of Search Results. In *SIGIR*. 10 pages. https://doi.org/10.1145/2348283.2348396
[51] Shuting Wang, Zhicheng Dou, Jiongnan Liu, Qiannan Zhu, and Ji-Rong Wen. 2024. Personalized and Diversified: Ranking Search Results in an Integrated Way. *ACM Trans. Inf. Syst.* 42, 3, Article 81 (Jan. 2024), 25 pages. https://doi.org/10.1145/3631989
[52] Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. 2016. Learning to Rank with Selection Bias in Personal Search. In *SIGIR*. 10 pages.

https://doi.org/10.1145/2911451.2911537

[53] Nishant Yadav, Nicholas Monath, Rico Angell, Manzil Zaheer, and Andrew McCallum. 2022. Efficient Nearest Neighbor Search for Cross-Encoder Models using Matrix Factorization. In *EMNLP*. https://aclanthology.org/2022.emnlp-main.140

[54] Le Yan, Zhen Qin, Xuanhui Wang, Michael Bendersky, and Marc Najork. 2022. Scale Calibration of Deep Ranking Models. In *KDD*. 10 pages. https://doi.org/10.1145/3534678.3539072

[55] Longqi Yang, Michael Sobolev, Yu Wang, Jenny Chen, Drew Dunne, Christina Tsangouri, Nicola Dell, Mor Naaman, and Deborah Estrin. 2019. How Intention Informed Recommendations Modulate Choices: A Field Study of Spoken Word Content. In *The Web Conference*. 12 pages. https://doi.org/10.1145/3308558.3313540

[56] Jing Yao, Zhicheng Dou, and Ji-Rong Wen. 2020. Employing Personal Word Embeddings for Personalized Search. In *SIGIR*. 10 pages. https://doi.org/10.1145/3397271.3401153

[57] Lucia Yu, Ethan Benjamin, Congzhe Su, Yinlin Fu, Jon Eskreis-Winkler, Xiaoting Zhao, and Diane Hu. 2021. Real-Time Personalized Ranking in E-commerce Search. *ECommerce Workshop at SIGIR* (2021).

[58] Hamed Zamani and Michael Bendersky. 2023. Multivariate Representation Learning for Information Retrieval. In *SIGIR*. 11 pages. https://doi.org/10.1145/3539618.3591740

[59] Hamed Zamani, Michael Bendersky, Xuanhui Wang, and Mingyang Zhang. 2017. Situational Context for Ranking in Personal Search. In *WWW*. 10 pages. https://doi.org/10.1145/3038912.3052648

[60] Binyam A. Zemede and Byron J. Gao. 2017. Personalized search with editable profiles. In *2017 IEEE International Conference on Big Data (Big Data)*. https://doi.org/10.1109/BigData.2017.8258572

[61] Hansi Zeng, Surya Kallumadi, Zaid Alibadi, Rodrigo Nogueira, and Hamed Zamani. 2023. A Personalized Dense Retrieval Framework for Unified Information Access. In *SIGIR*. 10 pages. https://doi.org/10.1145/3539618.3591626

[62] Yujia Zhou, Zhicheng Dou, Yutao Zhu, and Ji-Rong Wen. 2021. PSSL: Self-Supervised Learning for Personalized Search with Contrastive Sampling. In *CIKM*. 10 pages. https://doi.org/10.1145/3459637.3482379