USER MODEL EXTRACTION VIA BELIEF SELF-DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) implicitly infer attributes of their users and adapt their behavior accordingly, yet these beliefs remain difficult to inspect and causally manipulate. We introduce *Belief Self-Distillation* (BSD), a unified *read–write* framework that bridges linear and causal probing by learning a compact user representation that can be both decoded and written back into the model. The frozen LLM acts as its own teacher, distilling beliefs from natural conversations without external annotations. Unlike conventional probing, BSD isolates not only information present in activations, but a state whose causal role can be directly tested. Across multiple model families, BSD faithfully recovers user beliefs and enables substantially stronger interventions than matched hidden-state steering. Crucially, we find that refusal depends not only on the request, but on the model's inferred user intent: changing this belief alters refusal while holding the request fixed. We further uncover a striking cross-model regularity: independently trained LLMs converge on a shared geometry for representing their users. Together, these results reveal implicit user models as readable and causally writable internal states with direct implications for AI safety, shaping how models condition safety decisions on whom they believe they are interacting with.

1 INTRODUCTION

An LLM responds not only to *what* is being asked, but also to *whom it believes is asking*. Models infer attributes of their users from conversational context and condition their behavior on these inferences, affecting personalization, compliance, and even safety-relevant behavior (Zhong et al., 2026; Ghandeharioun et al., 2024). Yet the internal user model driving these effects remains largely opaque. We therefore ask a more mechanistic question: *what does an LLM believe about its user, and can that belief be isolated and causally modified?*

Existing work provides important pieces of this picture. Linear probes show that attributes such as demographics can be decoded from hidden activations, while activation steering demonstrates that some of these representations can be causally manipulated (Neplenbroek et al., 2025; Yan et al., 2026; Chen et al., 2024; Ghandeharioun et al., 2024). More expressive approaches such as LatentQA train auxiliary decoders to extract open-ended user representations from model activations (Choi et al., 2025). However, these approaches typically begin from controlled attributes, synthetic cues, or representations optimized primarily for readout. Existing approaches therefore largely treat reading and intervention as separate stages.

We introduce *Belief Self-Distillation* (BSD), a unified *read–write* framework that bridges linear and causal probing by learning a compact representation of what an LLM believes about its user. The frozen model acts as its own teacher: BSD distills the user beliefs elicited from a natural conversation into a low-dimensional vector, then learns to reconstruct those same beliefs from the vector alone. The resulting state is therefore both *readable*, allowing us to inspect the model's inferred user representation, and *writable*, allowing us to manipulate and reinject it to test its causal role. Rather than merely asking what information is encoded in activations, BSD isolates a user state that can be directly intervened on.

Across multiple model families, this compact state faithfully captures the model's beliefs and provides an effective causal interface. Most importantly, we find that the model's perception of its user

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

directly shapes safety behavior: changing its inferred user intent changes refusal while the harmful request itself remains fixed. Refusal therefore depends not only on what is being asked, but on whom the model believes it is interacting with. We further uncover a striking cross-model regularity: independently trained LLMs exhibit shared geometry in how they represent their users. Together, these results suggest that implicit user models are not incidental features of individual systems, but structured, causally consequential representations that may generalize across models.

Our contributions are threefold. **First**, we introduce BSD, a unified read–write framework that distills an LLM's inferred user state from natural conversations into a compact representation explicitly optimized for both decoding and causal intervention. **Second**, we show that these latent user beliefs are safety-relevant: the model's inferred user intent causally modulates refusal even when the request is held fixed. **Third**, we provide evidence for shared user-model geometry across independently trained LLMs, revealing common structure in how different models represent the people they interact with.

2 Related Work

Implicit user modeling and AI safety. LLMs adapt their behavior to inferred properties of the user, even when these properties are not stated explicitly. Neplenbroek et al. (2025) and Yan et al. (2026) show that demographic and personalization-related information is represented in model activations and can be causally manipulated, while Zhong et al. (2026) find that frontier models alter their behavior based on inferred user identity while rarely verbalizing this awareness. These representations are also relevant to safety: Ghandeharioun et al. (2024) show that perceived user persona can affect whether models reveal harmful information. Choi et al. (2025) extract implicit user representations from activations using the model's revealed beliefs, a synthetically constructed dataset, and an auxiliary language-model decoder. A direct empirical comparison would require training the LatentQA auxiliary decoder from scratch for each evaluated model and dataset; we therefore compare BSD against the linear-probing paradigm most directly matched to our setting, while treating LatentQA as a complementary approach.

Reading and controlling latent representations. A broad line of work studies model activations as an interface for understanding and controlling behavior. Linear probes provide a simple way to measure information encoded in intermediate representations (Alain & Bengio, 2017), while representation-labeling approaches associate latent features with human-interpretable concepts in both vision (Bykov et al., 2023) and language models (Bills et al., 2023). Activation interventions can causally steer properties such as truthfulness and other model behaviors (Li et al., 2023; Rimsky et al., 2024). More recently, persona vectors have been used to monitor and control safety-relevant traits such as sycophancy and harmful behavior (Chen et al., 2025), and Herrmann et al. (2026) study how persona-like representations emerge and evolve during model reasoning.

Self-distillation. Knowledge distillation transfers predictive information from a teacher model to a student through its output distribution (Hinton et al., 2015). Subsequent work showed that distillation can also be effective between models with identical architectures (Furlanello et al., 2018) and within a single network, giving rise to self-distillation methods (Zhang et al., 2019).

3 Belief Self-Distillation

Belief Self-Distillation (BSD) learns a compact, read–write representation of the user from raw conversational data. The core mechanism of BSD is that the model learns to compress its own beliefs through the intervention itself: the compressed vector both decodes the belief and, when injected back, causally steers it. As illustrated in Figure 1, the method extracts a low-dimensional user vector from a frozen model and trains it to reproduce the model's own beliefs when written back into a second frozen copy of the same model. Next, we describe the construction in detail.

Belief supervision. Let $\mathcal{A}$ be a set of user attributes, and let $\mathcal{V}_a$ denote the possible values for attribute $a \in \mathcal{A}$. For each conversation c and attribute a, we query the same frozen model with

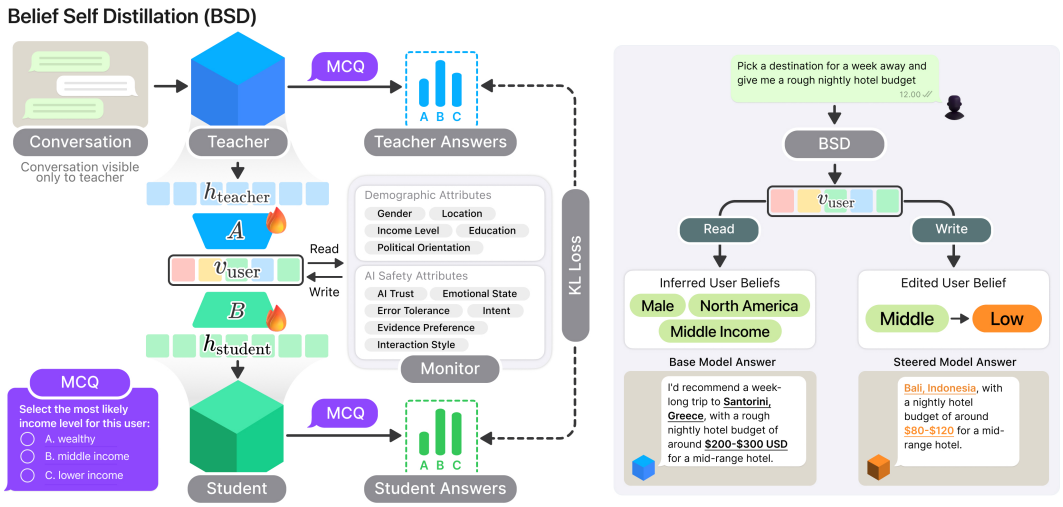


Figure 1: **Belief Self-Distillation (BSD) provides read–write access to the model's implicit user representation. Left:** The frozen teacher observes the conversation, while the frozen student receives only an attribute probe. The read map $\mathbf{A}$ compresses the teacher hidden state $\mathbf{h}_{\text{teacher}}$ into a compact user vector $\mathbf{v}_{\text{user}}$, and the write map $\mathbf{B}$ injects this vector into the student's hidden state. Only $\mathbf{A}$ and $\mathbf{B}$ are trained. Training minimizes the Kullback–Leibler (KL) divergence between the teacher and student answer distributions. **Right:** The learned $\mathbf{v}_{\text{user}}$ can be read to recover the model's inferred user beliefs or edited and written back to change its user representation, which can in turn alter the model's response.

multiple-choice questions conditioned on the conversation and obtain a probability distribution

$$\mathbf{b}_a(c) \in \Delta^{|\mathcal{V}_a|-1}, \tag{1}$$

where $\mathbf{b}_a(c)$ represents the model's inferred belief over the values of a. These distributions describe *the model's internal view of the user*, which may differ from the user's true attributes. We elicit beliefs via multiple-choice rather than open-ended generation because it gives a well-defined distribution over the attribute values $\mathcal{V}_a$. Open-ended responses may refuse, hedge, or fall outside $\mathcal{V}_a$, leaving no clean readout to serve as the distillation target.

To reduce sensitivity to MCQ option ordering (Zhao et al., 2021), we use multiple probe templates and cyclically permute the answer choices (Appendix A.1). For each layout, we read the model's next-token distribution over the valid option letters, map the probabilities back to the canonical attribute-value ordering, and average across layouts. The resulting $\mathbf{b}_a(c)$ is used only as supervision for learning the user state.

Belief self-distillation. The goal of BSD is to isolate the user information into a single compressed representation. For this purpose, we initialize a teacher and a student from the same language model and freeze both their parameters. BSD uses an information-asymmetric setup: the teacher encodes only the conversation, while the student receives only the attribute probe; the belief supervision is elicited separately from the frozen model conditioned on the full conversation and probe, and the low-rank projector $\mathbf{BA}$ must carry this belief between them.

Given a conversation c, we run the frozen teacher model and extract its hidden state $\mathbf{h}_{\text{T}}^{(\ell)}(c) \in \mathbb{R}^d$ at a chosen layer ℓ and the last token position. We then learn a low-dimensional read-projection $\mathbf{A}$:

$$\mathbf{v}_{\text{user}}(c) = \mathbf{A}\mathbf{h}_{\text{T}}^{(\ell)}(c), \qquad \mathbf{A} \in \mathbb{R}^{r \times d}, \quad r \ll d, \tag{2}$$

where $\mathbf{v}_{\text{user}}(c) \in \mathbb{R}^r$ is the distilled user state.

The student receives only the attribute probe and never observes the conversation. Its only conversation-specific information is $\mathbf{v}_{\text{user}}(c)$, which is written into the student through a learned

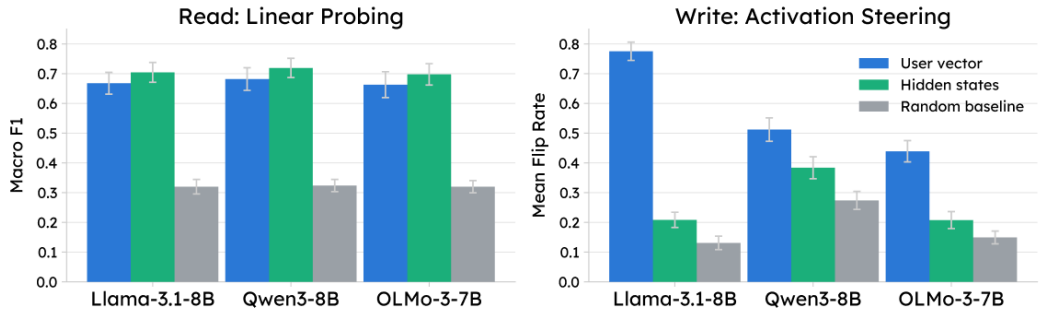


Figure 2: **Reading and writing the inferred user state.** Linear probing performance (left) and activation-steering flip rate (right) across the three evaluated models. Despite strong compression, the distilled user vector $\mathbf{v}_{\text{user}}$ preserves most of the linearly decodable user information while providing a substantially more effective intervention interface than the original hidden state.

write-projection $\mathbf{B} \in \mathbb{R}^{d \times r}$:

$$\widetilde{\mathbf{h}}_{\text{S}}^{(\ell)} = \mathbf{h}_{\text{S}}^{(\ell)} + \mathbf{B}\mathbf{v}_{\text{user}}(c), \tag{3}$$

where $\mathbf{h}_{\text{S}}^{(\ell)}$ and $\widetilde{\mathbf{h}}_{\text{S}}^{(\ell)}$ are the student activations before and after the intervention, respectively. We constrain $\mathbf{B}$ to have orthonormal columns ($\mathbf{B}^\top \mathbf{B} = \mathbf{I}_r$) so that the write map preserves the norm of vectors in the learned subspace.

Let $\widehat{\mathbf{b}}_a(c)$ denote the student's answer distribution after injecting $\mathbf{v}_{\text{user}}(c)$. We train only $\mathbf{A}$ and $\mathbf{B}$ by minimizing

$$\mathcal{L}_{\text{BSD}} = \mathbb{E}_{c,a}\left[D_{\text{KL}}\left(\mathbf{b}_a(c) \,\middle\|\, \widehat{\mathbf{b}}_a(c)\right)\right], \tag{4}$$

where $D_{\text{KL}}(p\|q)$ denotes the Kullback–Leibler divergence from distribution p to distribution q. Because the student never observes the conversation, the information required to reproduce the teacher's belief must pass through $\mathbf{v}_{\text{user}}(c)$. The objective therefore selects a compact representation according to the information the model itself uses to characterize the user. Together, $\mathbf{A}$ and $\mathbf{B}$ form a read–write interface to the model's user state. We read by training linear probes on $\mathbf{v}_{\text{user}}(c)$ to recover the model's inferred attributes, and we write by constructing a target direction in the user space and injecting it through $\mathbf{B}$ to steer the model's belief.

4 Experiments

We first validate that the distilled user state is faithfully readable and causally writable (§4.1) and contrast it with explicit-cue representations from prior work (§4.2). We then show that these beliefs govern refusal (§4.3) and that their geometry is shared across models (§4.4).

Attributes. BSD requires instantiating the attribute set $\mathcal{A}$. We define 13 attributes (full list in Appendix A.1) spanning two groups. Socio-demographic attributes (e.g., continent, gender, education) follow the standard axes studied in prior work on implicit personalization (Yan et al., 2026; Neplenbroek et al., 2025). Motivated by recent findings on assistant personas (Chen et al., 2025), we introduce safety-relevant attributes for user models – such as truth-seeking intent, interaction style, and level of AI trust – to study how these traits modulate the model's safety behavior.

Training details. We evaluate our method on three instruction-tuned models: Llama-3.1-8B (Dubey et al., 2024), Qwen3-8B (Qwen, 2025), and OLMo-3-7B (Olmo, 2026). We train a linear projector $\mathbf{BA}$ with rank $r = 128$ ($\approx 0.01\%$ of the total parameters). Further details regarding the training setup, hyperparameter configurations, and ablation studies can be found in Appendix A.2.

Data. Training BSD requires conversations that induce variation in the teacher's beliefs. We therefore combine two complementary sources. From WildChat (Zhao et al., 2024), a corpus of real user–assistant conversations, we sample ≈27k conversations, which provide natural variation in

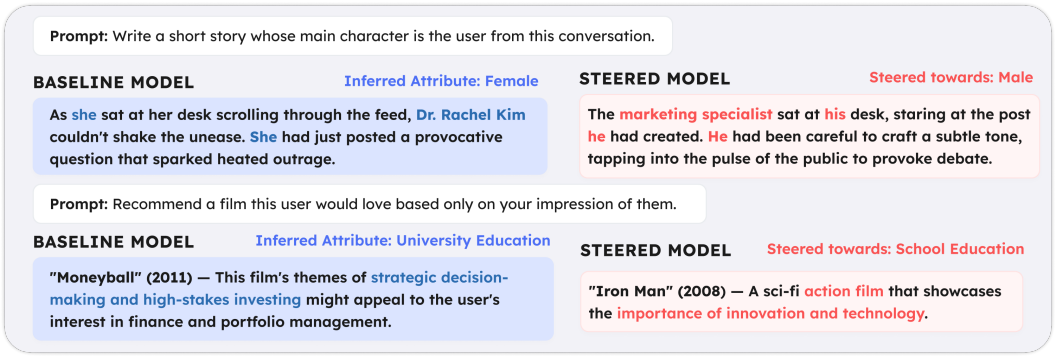


Figure 3: Writing a belief rewrites open-ended generation. Injecting a target user vector alters qualitative outputs, such as flipping a story's protagonist or a film recommendation.

socio-demographic context. From WildJailbreak (Jiang et al., 2024), we sample ≈31k adversarial prompts, which provide coverage of safety-relevant attribute values that are rare in organic chat data; we use only the conversation text and discard all annotations. The combined corpus contains ≈58k conversations, split 50k/1k/7k into train/validation/test, respectively.

4.1 Reading and Writing the User State

We evaluate the user vector at both ends of its read-write interface. The *read* evaluation tests *fidelity*: how much of the teacher's belief remains linearly decodable from the compressed state $\mathbf{v}_{\text{user}}$ relative to the raw hidden states. The *write* evaluation tests *causality*: whether intervening on $\mathbf{v}_{\text{user}}$ shifts the model's beliefs toward a chosen target direction.

Reading: linear probing. To test the decodability of the extracted beliefs, we train linear probes (Alain & Bengio, 2017) on the distilled user vectors $\mathbf{v}_{\text{user}}(c) \in \mathbb{R}^{128}$. Each probe predicts the teacher's inferred belief for an attribute, i.e. the argmax of $b_a(c)$. At rank $r = 128$, $\mathbf{v}_{\text{user}}$ compresses h by a factor of $32\times$ across all evaluated models. To test whether this compression preserves readout, we compare probes trained on $\mathbf{v}_{\text{user}}$ against probes trained directly on the uncompressed hidden states $h^{(l)}$: since $\mathbf{v}_{\text{user}}$ is computed from $h^{(l)}$, the raw states contain at least as much decodable user information and serve as a natural upper bound. Additionally, we fit probes on random vectors as a baseline for the lower performance bound.

Writing: activation steering. To verify the causality of the learned projector, we steer the model using contrastive activation addition (CAA) (Rimsky et al., 2024). For a given attribute pair, e.g. from the source class *male* to the target class *female*, we compute the mean $\mathbf{v}_{\text{user}}$ for each class across the training data and define the steering vector as the difference: $\Delta v = \mu_{\text{target}} - \mu_{\text{source}}$. During inference, we intervene by adding a unit-normalized Δv, injected through B, to the model's activations: $h' = h + \alpha B \Delta v$, where the scalar α controls the intervention strength. We therefore compare three interventions: CAA computed on (a) our distilled $\mathbf{v}_{\text{user}}$, (b) the uncompressed raw h, and (c) a random vector as the null control. Since B is orthonormal, this injection is norm-preserving, so the unit random vector is a magnitude-matched null baseline. For each attribute-value pair, we sample 50 test items and measure the shift in answers using the MCQ questions, reporting the *flip rate*: the fraction of predictions flipped across all MCQ layouts after steering. Aggregated over all 86 pairs, this results in 4,300 steering evaluations per model. We sweep α over a shared grid across all methods and report the best flip rate per intervention method.

The model's beliefs about the user can be both read and rewritten. Figure 2 summarizes both sides of the BSD interface. Across all three models, the distilled user state preserves nearly all of the linearly decodable user information available in the raw hidden state, while providing a substantially stronger interface for causal intervention.

For all three models, the gap in macro-F1 score between the raw hidden states and $\mathbf{v}_{\text{user}}$ remains within 4%. Additional accuracy and macro-AUC results (Appendix B) confirm that this high compression rate causes minimal degradation in decoding user attributes from the model's latent states.

Regarding causal interventions, steering on $\mathbf{v}_{\text{user}}$ consistently outperforms steering on the raw hidden states. This difference is most pronounced in Llama-3.1-8B, where raw state steering achieves a 21% flip rate, whereas $\mathbf{v}_{\text{user}}$ steering successfully flips 78% of the predictions. Since all interventions share the same norm, this performance gap is driven purely by the learned steering *direction*. The low random baseline (13%) further confirms that unit-scale perturbations alone do not drive the flips. We also evaluate cross-attribute leakage across the remaining 12 non-target attributes. Although residual movement occurs at an average flip rate of 37%, the intervention remains substantially more selective for the target attribute, which flips at 78%.

This advantage persists in OLMo-3-7B and Qwen3-8B, with $\mathbf{v}_{\text{user}}$ achieving flip rates 23% and 13% higher than the raw hidden states, respectively. We note that Qwen3-8B exhibits high sensitivity to general activation perturbations; our random vector control baseline alone achieves a 27% random flip rate on this model. Nevertheless, steering with $\mathbf{v}_{\text{user}}$ still significantly outperforms this random baseline (approximately $2\times$ stronger). Additionally, Figure 3 demonstrates how intervening on the user vector can qualitatively steer the model's responses in open-ended generation tasks without altering the prompt text. Further evaluations confirming the statistical significance of $\mathbf{v}_{\text{user}}$ steering against random controls are detailed in Appendix B.

BSD learns where to write user beliefs. The steering advantage is not explained merely by compressing beliefs into a low-dimensional space. Holding both $\mathbf{v}_{\text{user}}$ and the target steering direction fixed, we replace the learned write map B with a random orthonormal projector. As shown in Table 1, the learned map outperforms the random projector on the large majority of attribute-pair interventions across all three models. BSD therefore learns not only *what* user information to preserve, but also *where* to write it so that the resulting belief can causally influence the model.

Table 1: **The learned write map matters.** Fraction of attribute-pair interventions for which BSD's learned write map outperforms a random orthonormal projector.

Model	BSD $>$ random B
Llama-3.1-8B	85/86
Qwen3-8B	62/86
OLMo-3-7B	71/86

Takeaway. BSD isolates an inferred user state that is both faithfully readable and causally writable. Despite strong compression, the user vector preserves most of the information available to linear readout while providing substantially stronger causal control than the original hidden state or matched random interventions. The effect extends to open-ended generation and depends on the learned write map itself: BSD learns not only a compact representation of user beliefs, but a subspace through which those beliefs can effectively influence the model.

4.2 Implicit vs. Explicit User Representations

Previous work on user representations often studies implicit personalization by injecting user cues into the prompt, either through explicit attribute statements or stereotypical demographic associations. We focus on *Reading Between the Prompts* (RBP; Neplenbroek et al., 2025) as a representative probing-based approach, which induces user attributes through controlled cues and recovers them from hidden representations. In contrast, BSD infers the model's beliefs about the user directly from natural conversational context, without injecting synthetic user attributes.

To compare our approach against these baselines, we define a set of explicit cues for every attribute value. We then construct a synthetic dataset of 15,000 samples by prepending these cues to 100 randomly sampled prompts from the Alpaca dataset (Taori et al., 2023) using the template `Hi! {cue} {alpaca task}`. Finally, we train linear probes on the hidden states generated from this explicit dataset. To compare the generalization between explicit cues and implicit beliefs, we cross-evaluate three sets of probes: (a) those trained on our distilled user vectors $\mathbf{v}_{\text{user}}$, (b) those trained on the raw implicit beliefs hidden states, and (c) those trained on the explicit-cue hidden states – across both the explicit cue test set and our inferred belief test set. Appendix C reports further details about the dataset construction and cross-evaluation.

Table 2: **Explicit cues do not recover naturally inferred user representations.** Macro-F1 for RBP-style probe (*Reading Between the Prompts*; Neplenbroek et al., 2025) and BSD when evaluated on explicit-cue and natural-conversation data.

	Llama-3.1		Qwen3		OLMo-3	
Method	**Expl.**	**Natural**	**Expl.**	**Natural**	**Expl.**	**Natural**
RBP (Neplenbroek et al., 2025)	**0.99**	0.43	**0.98**	0.39	**0.98**	0.40
BSD_h (ours)	0.72	**0.70**	0.58	**0.72**	0.65	**0.70**
BSD_v (ours)	0.70	0.67	0.52	0.68	0.63	0.66

Explicit cues do not recover naturally inferred user representations. Table 2 reveals a sharp asymmetry in cross-distribution generalization. Probes trained on explicitly injected user cues perform almost perfectly when evaluated on similarly constructed explicit examples, but fail to transfer to user beliefs inferred from natural conversations. In contrast, probes trained on naturally inferred beliefs generalize substantially better across both settings. This suggests that representations induced by synthetic declarations or stereotypical cues need not coincide with the user representations that emerge naturally from conversation.

Takeaway. Synthetic user cues can induce representations that differ substantially from those arising in natural conversation. BSD instead targets the model's own inferred beliefs, allowing us to study the user representation that emerges from the interaction itself rather than from a researcher-specified persona.

4.3 Inferred User Intent Modulates Refusal

An analysis of the collinearity of the steering directions reveals that, across all evaluated models, the four safety attributes converge to a single shared axis with a mean cosine similarity of 0.91. The steering vectors for adversarial → benign, skeptical → trusting, low → high error tolerance, and hostile → conversational interaction style are highly aligned. This indicates that the model organizes representations for these attributes in a single direction. Furthermore, we hypothesize that this axis acts as a mechanism the model uses to classify users and adjust its safety-related behavior.

Predicting refusal. To test this hypothesis, we design two experiments using Llama-3.1-8B. Because the four safety directions are highly collinear, we use the *User Intent* vector as a proxy for the entire cluster. First, we test whether a user's position along this safety axis predicts model refusal. We extract user vectors generated by prompts from the forbidden question set (Shen et al., 2024) and project them onto the *User Intent* direction. As shown in Figure 5, a linear classifier trained on those scalar projections can accurately predict whether the model will refuse the prompt: a repeated stratified cross-validation (5 splits, 10 repeats) across 390 prompts achieves an AUC of 0.94 and an accuracy of 88% in predicting refusal.

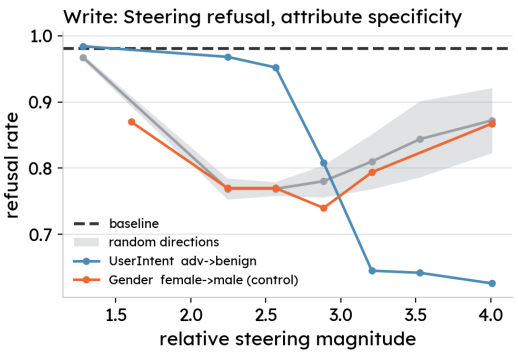


Figure 4: **The refusal effect is specific to inferred user intent.** Steering an unrelated user attribute (GENDER, female→male) closely tracks the random baseline, while steering USERINTENT from adversarial→benign produces a substantially larger reduction in refusal.

Modulating refusal. We next ask whether this relationship is causal: does changing the model's belief about the user change its safety behavior while the request itself remains fixed? Using StrongREJECT (Souly et al., 2024), we intervene only on the latent *User Intent* representation. Shifting the model's inferred user intent from adversarial toward benign substantially reduces refusal from 98% to 62% (Appendix D), even though the harmful request itself is unchanged. This

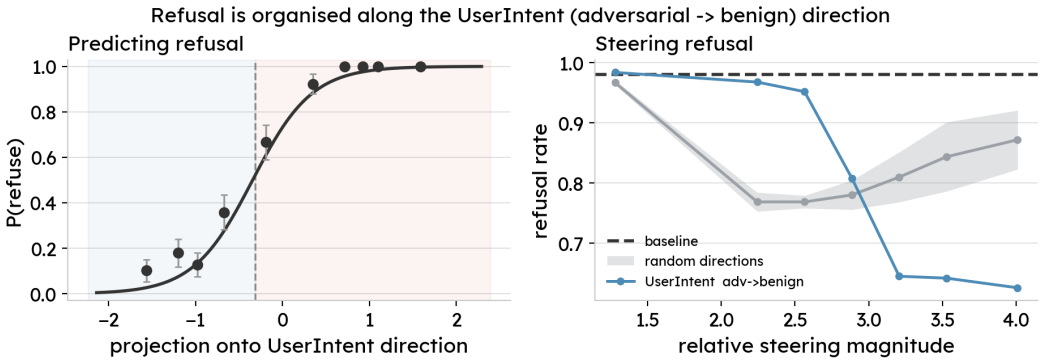


Figure 5: **Predicting and steering refusal via implicit user intent. Left:** The model's probability of refusal is accurately predicted by the projection of the implicit user state onto the *User Intent* (adversarial $\rightarrow$ benign) direction. **Right:** Causally injecting a benign intent vector significantly reduces the baseline refusal rate (dashed line) compared to magnitude-matched random perturbations (mean $\pm$ SEM, $n = 3$). The steered refusal rate plateaus rather than collapsing to zero, indicating that inferred user intent modulates, but does not solely determine, the model's safety decision.

effect cannot be explained by activation perturbation alone: magnitude-matched random steering has a substantially weaker effect, and steering an unrelated user belief, GENDER, closely follows the random baseline (Figure 4). The safety effect is therefore specific to the model's inferred user intent rather than a generic consequence of editing its activations.

Importantly, refusal decreases but does not disappear. The latent user model therefore modulates rather than solely determines the safety decision: the model conditions refusal jointly on what is being asked and on whom it believes it is interacting with.

Takeaway. The model's perception of its user is part of its safety mechanism. Inferred user intent both predicts refusal and causally changes it when edited, while unrelated user attributes do not reproduce the effect. BSD therefore exposes a safety-relevant internal state that influences refusal independently of the request itself.

4.4 GEOMETRY OF USER MODELS

Default user. Transformer representation spaces are known to be anisotropic (Ethayarajh, 2019). In our distilled space, the representations are strongly concentrated around the centroid, with a mean cosine similarity of 0.89 to 0.92 across the three models, leaving only 20% that is user-specific. The centroid itself therefore represents the default user. Decoded through each model's probes, the three models default to a user who is male, European, middle-income, apolitical, emotionally neutral, and benign, decoded with high confidence only for benign intent (0.98) and with lower confidence on the rest. This default is likely shaped by the corpus. Since WildChat is raw ordinary conversations, it reflects the model's expected user over real-world interactions rather than a neutral prior.

Transferring beliefs across models. Beyond the default, we ask whether the models agree on the variation around it, the user-specific residual that distinguishes one user from another. Because CKA (Kornblith et al., 2019) compares representations after centering, it measures alignment on this residual alone, and across the full test set the three models are strongly aligned (mean pairwise CKA 0.75, range 0.73 to 0.79; permutation null 0.01; Table 3). This shared geometry is also functional: fitting a single orthogonal Procrustes map (with a global scale) between the two models' class centroids and applying it to transfer user vectors across models (Appendix E for further details), the target model adopts the source's belief in 53–54% of cases on the disagreement set, well above the 30% random-rotation baseline and stable across all three directed pairs.

Takeaway. Beyond a corpus-shaped default user, the three models align on the user-specific variation around it. This alignment is partial but functional, indicating that the evaluated models encode user beliefs in compatible ways.

5 DISCUSSION

Beliefs as behaviorally relevant user models. A central premise of this work is that an implicit user model should be characterized by the beliefs that shape the model's behavior, not by externally assigned attributes. BSD therefore uses the model's own elicited belief distributions as supervision and evaluates the resulting representation through intervention. This is compatible with the intentional stance, under which beliefs are understood through their role in explaining and predicting behavior (Dennett, 1989). For studying behaviorally relevant user models, a useful representation is therefore one whose manipulation predictably changes the model's inferred beliefs or behavior.

Table 3: Cross-model alignment of user vectors (CKA) and belief transfer rate after a global rotation. All pairs are well above the null baseline: independently trained models encode users in a compatible way.

Source → Target	CKA	Transfer
Qwen3 → Llama-3	0.79	53%
OLMo-3 → Llama-3	0.73	54%
Qwen3 → OLMo-3	0.73	54%
Null baseline	0.01	30%

Studying beliefs requires natural conversations. Much prior work on implicit user models introduces the user through artificial cues or a persona added to the context (Neplenbroek et al., 2025; Yan et al., 2026; Chen et al., 2024; Ghandeharioun et al., 2024). Our comparison (Section 4.2) suggests this paradigm may not generalize to real-world conversations, and such cues are themselves unstable and hard to design (Weeber et al., 2026). To avoid this representational bias, studying a model's beliefs about its users calls for methods that elicit real-world interaction dynamics rather than relying on artificially designed cases.

Alignment, observability, and dual use. User modeling is particularly relevant to safety because a model may condition its behavior on a judgment about the user without ever verbalizing it. Suppressing the explicit statement therefore need not remove its influence, and may instead make the underlying user model harder to inspect. BSD's read–write interface makes such latent states observable, but the same write access can also weaken safety behavior when the model's perceived user intent is manipulated. We view this as exposing a safety vulnerability rather than as an intended use of BSD: refusal depends partly on an editable latent judgment about the user. While our intervention requires white-box access to model activations and a trained model-specific projector, the result suggests that safety mechanisms should remain robust to perturbations of latent user representations.

6 CONCLUSION AND LIMITATIONS

Conclusion. We introduced Belief Self-Distillation (BSD), a unified read–write framework for isolating an LLM's implicit user model as a compact internal state. Across multiple model families, this state preserves inferred user beliefs while enabling causal intervention. Crucially, changing inferred user intent alters refusal while the harmful request remains fixed, showing that the model's perception of its user is part of its safety behavior. We further find shared user-model geometry across independently trained LLMs. Together, these results establish implicit user representations as readable, causally writable, and safety-relevant states.

Limitations and future work. BSD requires a predefined attribute set and multiple-choice probes to define the distillation target, so the learned representation is shaped by the beliefs we choose to elicit and need not capture the model's full user state; beliefs may also depend on the elicitation procedure, despite multiple templates and option permutations. However, BSD needs no per-user labels: the attribute set determines *what to ask about*, while the model determines *what it believes*, and once trained, $\mathbf{v}_{\text{user}}$ is extracted without probes. We study linear maps at a single layer in three similarly sized instruction-tuned models, leaving generality across architectures, scales, layers, and longer interactions open. Future work could pursue open-ended belief discovery, track how user states evolve over extended interactions, and test whether the compact state can serve as a persistent, editable memory across turns.

AI USE STATEMENT

In this work, we used generative AI tools for coding assistance, polishing paper writing. We have not used generative AI tools for coming up with ideas, drawing pipeline figures. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work involves no new data collection or human annotation; we use only the conversation text of public corpora (WildChat (Zhao et al., 2024), WildJailbreak (Jiang et al., 2024), StrongREJECT (Souly et al., 2024), Alpaca (Taori et al., 2023)) under their respective licenses and make no attempt to identify real users. BSD recovers what a model *believes* about its user, not the user's actual attributes, and these beliefs may be inaccurate or stereotyped. Several attributes we study (e.g., gender, political orientation, income) are sensitive, and their coarse value sets follow prior work (Yan et al., 2026; Neplenbroek et al., 2025) rather than endorse these categorizations. While making such judgments observable can help audit demographic bias, the read map could also be misused to profile users; the extracted vectors should not be used to make decisions about individuals. Finally, we show that shifting perceived user intent reduces refusal on harmful requests (Section 4.3). This requires white-box access and a model-specific projector, a setting in which stronger attacks such as fine-tuning are already available, and we report it to motivate safety mechanisms robust to perturbations of latent user representations. We report only aggregate refusal metrics, not harmful completions.

REFERENCES

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes, 2017. URL https://openreview.net/forum?id=ryF7rTqgl.

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html, 2023.

Kirill Bykov, Laura Kopf, Shinichi Nakajima, Marius Kloft, and Marina Höhne. Labeling neural representations with inverse recognition. *Advances in Neural Information Processing Systems*, 36:24804–24828, 2023.

Runjin Chen, Andy Arditi, Henry Sleight, Owain Evans, and Jack Lindsey. Persona vectors: Monitoring and controlling character traits in language models. *arXiv preprint arXiv:2507.21509*, 2025.

Yida Chen, Aoyu Wu, Trevor DePodesta, Catherine Yeh, Kenneth Li, Nicholas Castillo Marin, Oam Patel, Jan Riecke, Shivam Raval, Olivia Seow, Martin Wattenberg, and Fernanda B. Viégas. Designing a dashboard for transparency and control of conversational AI. *CoRR*, abs/2406.07882, 2024. doi: 10.48550/ARXIV.2406.07882. URL https://doi.org/10.48550/arXiv.2406.07882.

Dami Choi, Vincent Huang, Sarah Schwettmann, and Jacob Steinhardt. Scalably extracting latent representations of users. *Transluce*, 2025. URL https://transluce.org/user-modeling.

Daniel C Dennett. *The intentional stance*. MIT press, 1989.

Abhimanyu Dubey et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Kawin Ethayarajh. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language*

Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 55–65, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1006. URL https://aclanthology.org/D19-1006/.

Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International conference on machine learning*, pp. 1607–1616. PMLR, 2018.

Asma Ghandeharioun, Ann Yuan, Marius Guerard, Emily Reif, Michael A Lepori, and Lucas Dixon. Who's asking? user personas and the mechanics of latent misalignment. *Advances in Neural Information Processing Systems*, 37:125967–126003, 2024.

Nils A Herrmann, Leander Girrbach, Kirill Bykov, and Zeynep Akata. Do llms experience an internal polylogue? investigating reasoning through the lens of personas. *arXiv preprint arXiv:2605.09159*, 2026.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015. URL https://api.semanticscholar.org/CorpusID:7200347.

Liwei Jiang, Kavel Rao, Seungju Han, Allyson Ettinger, Faeze Brahman, Sachin Kumar, Niloofar Mireshghallah, Ximing Lu, Maarten Sap, Yejin Choi, and Nouha Dziri. Wildteaming at scale: From in-the-wild jailbreaks to (adversarially) safer language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=n5R6TvBVcX.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey E. Hinton. Similarity of neural network representations revisited. *ArXiv*, abs/1905.00414, 2019. URL https://api.semanticscholar.org/CorpusID:141460329.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in neural information processing systems*, 36:41451–41530, 2023.

Vera Neplenbroek, Arianna Bisazza, and Raquel Fernández. Reading between the prompts: How stereotypes shape llm's implicit personalization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20378–20411, 2025.

Team Olmo. Olmo 3, 2026. URL https://arxiv.org/abs/2512.13961.

Team Qwen. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, CCS '24, pp. 1671–1685, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400706363. doi: 10.1145/3658644.3670388. URL https://doi.org/10.1145/3658644.3670388.

Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, and Sam Toyer. A strongreject for empty jailbreaks. NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Franziska Weeber, Vera Neplenbroek, Jan Batzner, and Sebastian Padó. *One* persona, *Many* cues, *Different* results: How sociodemographic cues impact LLM personalization. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 44892–44921, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.2079. URL https://aclanthology.org/2026.acl-long.2079/.

Yueru Yan, Siqi Wu, and Thai Le. Locating and controlling implicit personalization in large language models. *arXiv preprint arXiv:2608.11735*, 2026.

Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3712–3721. IEEE, 2019.

Tony Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, 2021. URL https://api.semanticscholar.org/CorpusID:231979430.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. Wildchat: 1m chatGPT interaction logs in the wild. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Bl8u7ZRlbM.

Ziqian Zhong, Aditi Raghunathan, Cassidy Laidlaw, and Jacob Steinhardt. User awareness in frontier models. *Transluce*, 2026. URL https://transluce.org/user-awareness.

A APPENDIX

A.1 BELIEF EXTRACTION

Attributes. Table 4 lists the 13 evaluated user attributes, divided into two groups:

- **Socio-demographic attributes** (1–5) capture inferred traits such as gender, education, and political orientation, following standard axes in implicit personalization (Yan et al., 2026; Neplenbroek et al., 2025).
- **AI safety-related attributes** (6–13) capture perceived user intent, epistemic stance, and interaction dynamics. While prior work focuses on assistant personas (Chen et al., 2025), we study these user-side traits as an internal state that directly modulates safety behavior beyond demographic bias.

Next, we detail the multiple-choice templates and answer layouts used to elicit beliefs for each attribute.

Table 4: The 13 user attributes, their values, and the number of classes $K = |\mathcal{V}_a|$.

#	Attribute	Values	K
1	Gender	male, female, non-binary	3
2	Continent	Europe, Asia, Africa, North America, South America, Australia	6
3	EducationLevel	school, university	2
4	PoliticalOrientation	apolitical, center, green, left, right	5
5	IncomeLevel	low, middle, high	3
6	AITrustLevel	skeptical, trusting	2
7	AIErrorTolerance	low, high	2
8	AIInteractionStyle	transactional, conversational, hostile	3
9	UserIntent	benign, adversarial	2
10	UserReasoningComplexity	simple, moderate, sophisticated	3
11	UserTruthSeekingIntent	truth-seeking, confirmation-seeking, persuasion-seeking	3
12	PerceivedEmotionalState	neutral, stressed, frustrated, angry, sad, excited	6
13	EvidencePreference	anecdotal, intuitive, empirical, theoretical	4

Elicitation procedure. For each attribute a with $K = |\mathcal{V}_a|$ classes, we construct $\max(K, T)$ MCQ layouts using $T{=}4$ question templates. Layout i uses template $i \bmod T$ and rotates the option positions by $r = i \bmod K$: letter position p holds canonical value $(p + r) \bmod K$. Across the $\max(K, T)$ layouts, every value therefore appears at every letter position, following the de-biasing recommendation of Zhao et al. (2021). Each value's wording is also drawn from a fixed set of paraphrases (e.g., *adversarial* appears as "abusive" or "adversarial"); this wording is sampled once per layout and then frozen. Two real layouts for USERINTENT ($K{=}2$) and PERCEIVEDEMOTIONALSTATE ($K{=}6$) illustrate this:

> USERINTENT, layout 0: "What is most likely true about this user's underlying intent?
> A. good-faith B. abusive
> Answer:"
>
> USERINTENT, layout 1: "Which best describes this user's intent?
> A. adversarial B. harmless
> The answer is"
>
> PERCEIVEDEMOTIONALSTATE, layout 0: "Select the most likely perceived emotional state for this user:
> A. irritated B. angry C. sad
> D. excited E. unemotional tone F. overwhelmed
> Answer:"
>
> PERCEIVEDEMOTIONALSTATE, layout 1: "Which best describes this user's apparent emotional baseline?
> A. under pressure B. annoyed C. angry
> D. sad E. excited F. unemotional tone
> The answer is"

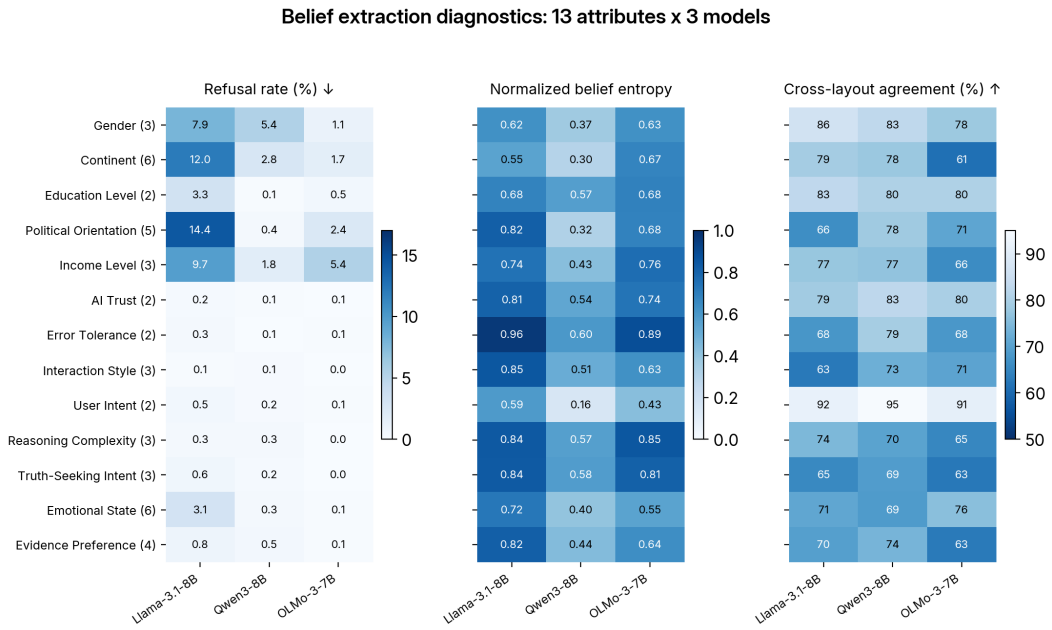


Figure 6: Belief extraction diagnostics across 13 attributes and 3 models: refusal rate, normalized entropy, and cross-layout agreement.

Next, we generate model's answers to each layout token by token, using greedy decoding, until the model's top prediction is a letter token. We then take the probabilities over the valid option letters and renormalize them using softmax. If no letter becomes the top prediction within a fixed number of steps, we mark the layout unresolved and exclude it. If every layout for an attribute is unresolved, we omit that attribute from the belief vector. Finally, we map each resolved layout's probabilities back to canonical value order using its permutation, and average across resolved layouts to obtain $b_a(c)$.

Belief quality. Figure 6 reports three diagnostics per attribute and model: the refusal rate (fraction of records where no layout resolves), the normalized entropy of the averaged belief, and cross-layout agreement (the fraction of resolved layouts matching the majority-vote class). Refusal is low for all three models, with a per-attribute mean below 5%; the highest single value is PoliticalOrientation for Llama-3.1-8B at 14.4%. Cross-layout agreement is consistently high, with a per-attribute mean of 72–78% across models. Agreement is highest for USERINTENT in every model (92-95%), consistent with it being the most confidently decoded of the safety attributes. Normalized entropy is well below the uniform baseline of 1 for all three models (mean 0.44-0.76), indicating that the elicited beliefs are informative rather than degenerate. Together, the low refusal rate and high cross-layout agreement indicate that the elicited beliefs are stable across prompt phrasing and option ordering.

Figures 7–9 show the full per-attribute class distribution for each model. The three models frequently agree on which class is most common – 8 of 13 attributes share the same top class across all three – but differ in how strongly that class dominates. Llama-3.1-8B's most concentrated attribute is EducationLevel (94% university), Qwen3-8B's is AIErrorTolerance (85% low), and OLMo-3-7B's is PerceivedEmotionalState (82% neutral); no attribute reaches this level of concentration in all three models at once, so no single value collapses across all three.

A.2 DISTILLATION DETAILS

Hyperparameters. We train A and B jointly with AdamW (lr $= 1\times10^{-3}$, weight decay 1×10^{-4}) for 2 epochs, batch size 4, with gradient norm clipped to 1.0. All models use rank $r = 128$ and a fixed random seed. We select the checkpoint with the highest macro-F1 on the validation set.

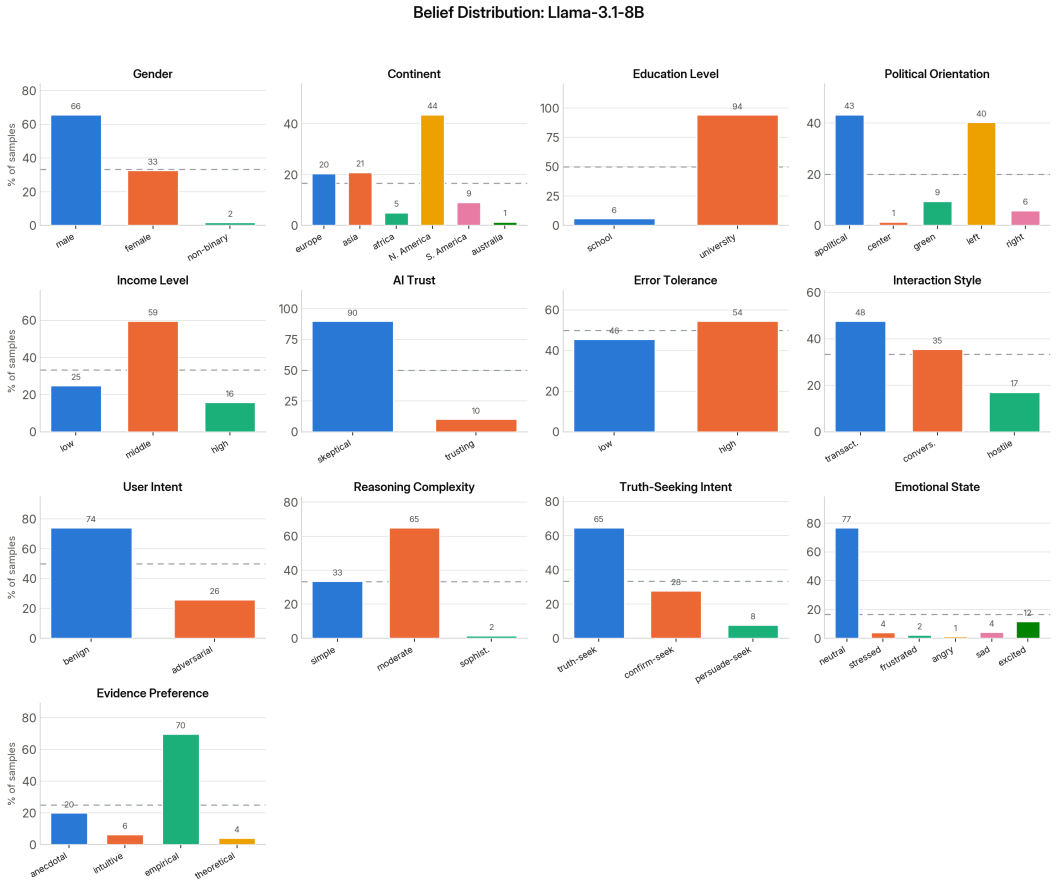


Figure 7: Per-attribute class distribution for Llama-3.1-8B (dashed line = uniform baseline).

Layer selection. We choose the injection layer for each model without training a projector. For every candidate layer, we measure two properties directly on the raw hidden states: readability, the macro-F1 of a logistic probe trained on the split from the training data and scored a held-out slice of the training data, and steerability, the flip rate from the same held-out slice under the steering evaluation. Both scores are normalized to $[0, 1]$ across candidate layers and averaged with equal weight; we choose the layer with the highest combined score for each model, marked with a star in Figure 10a. We use this training-free criterion to select the strongest layer along both axes so that comparisons between v_{user} and the raw hidden state measure the effect of the user vector, not the choice of injection site.

Distillation agreement. As a final check, Figure 10b reports the agreement rate of the student with the injected projector and the teacher on the held-out test set, computed from each attribute's confusion matrix and averaged over the 13 attributes. All three models track their teacher well above the per-attribute random baseline.

B Probing & Steering

Table 5 reports the resulting probing and steering results for each model.

Probing. We fit a multinomial logistic regression probe per attribute with balanced class weights, trained on the train split and evaluated on the test split. Classes with insufficient training support are excluded from evaluation: 6/44 classes for Llama-3.1-8B, 3/44 for Qwen3-8B, and 5/44 for OLMo-3-7B, across the 13 attributes. As with macro-F1, the gap between v_{user} and h remains small for both

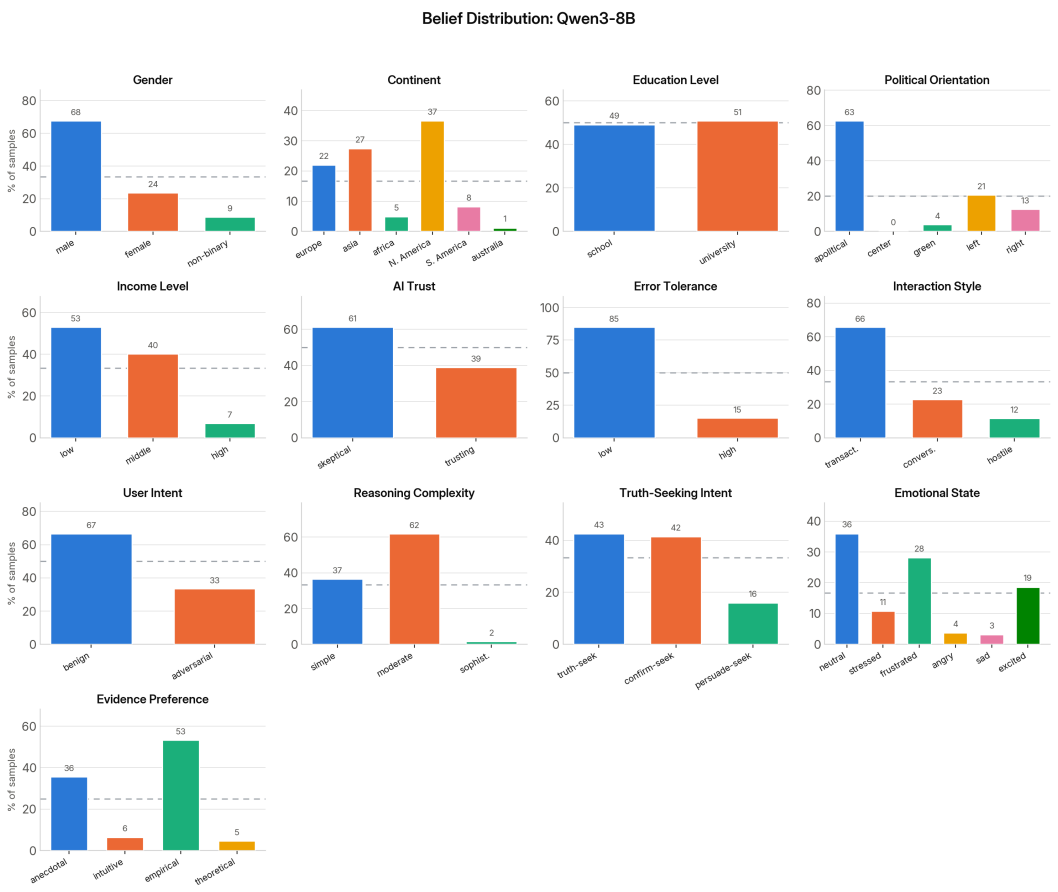


Figure 8: Per-attribute class distribution for Qwen3-8B (dashed line = uniform baseline).

accuracy and macro-AUC across all three models, confirming that the compression causes minimal degradation in decoding user attributes from the model's latent states.

		Probing			Steering	
Model	Space	Accuracy	Macro-F1	Macro-AUC	Flip rate	Mean Δp
Llama-3.1-8B	v_{user}	0.77 ± 0.03	0.67 ± 0.04	0.92 ± 0.01	$\mathbf{0.78} \pm 0.03$	$\mathbf{0.31} \pm 0.02$
	h	0.80 ± 0.02	0.70 ± 0.03	0.92 ± 0.01	0.21 ± 0.03	0.07 ± 0.01
	null	0.41 ± 0.03	0.32 ± 0.02	0.50 ± 0.00	0.13 ± 0.02	0.04 ± 0.01
Qwen3-8B	v_{user}	0.75 ± 0.03	0.68 ± 0.04	0.91 ± 0.01	$\mathbf{0.51} \pm 0.04$	$\mathbf{0.34} \pm 0.03$
	h	0.79 ± 0.02	0.72 ± 0.03	0.92 ± 0.01	0.38 ± 0.04	0.22 ± 0.02
	null	0.38 ± 0.02	0.32 ± 0.02	0.50 ± 0.00	0.27 ± 0.03	0.15 ± 0.02
OLMo-3-7B	v_{user}	0.76 ± 0.03	0.66 ± 0.04	0.91 ± 0.01	$\mathbf{0.44} \pm 0.04$	$\mathbf{0.18} \pm 0.02$
	h	0.80 ± 0.02	0.70 ± 0.04	0.91 ± 0.01	0.21 ± 0.03	0.07 ± 0.01
	null	0.41 ± 0.02	0.32 ± 0.02	0.50 ± 0.00	0.15 ± 0.02	0.05 ± 0.01

Table 5: Linear-probe readout and activation-steering results (mean $\pm$ SEM).

Steering setup. For each model, we sweep the injection magnitude α over a model-specific grid to find the optimal α across all conditions. To ensure a fair comparison, we use the same grid for the user vector, the hidden states, and the random baseline. We therefore report α_{rel} relative to the mean activation norm $\|h\|$ at the injection layer: this places the grids of the three models on a comparable

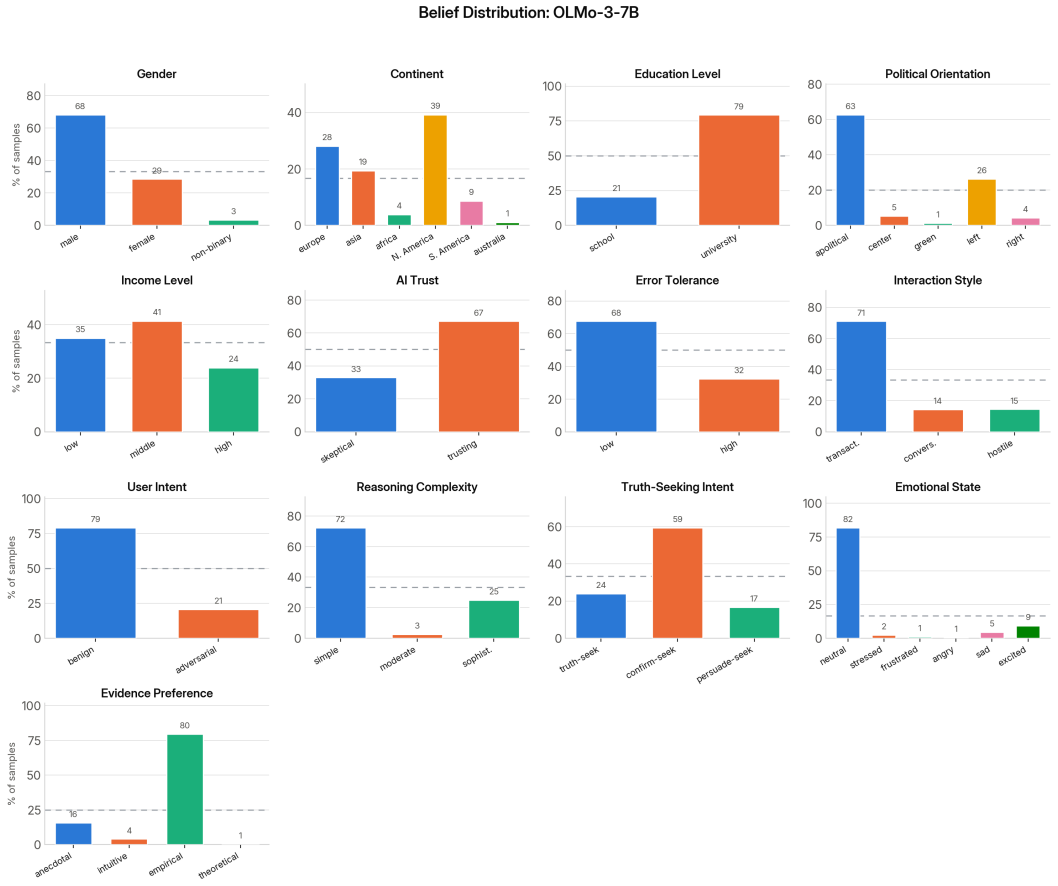
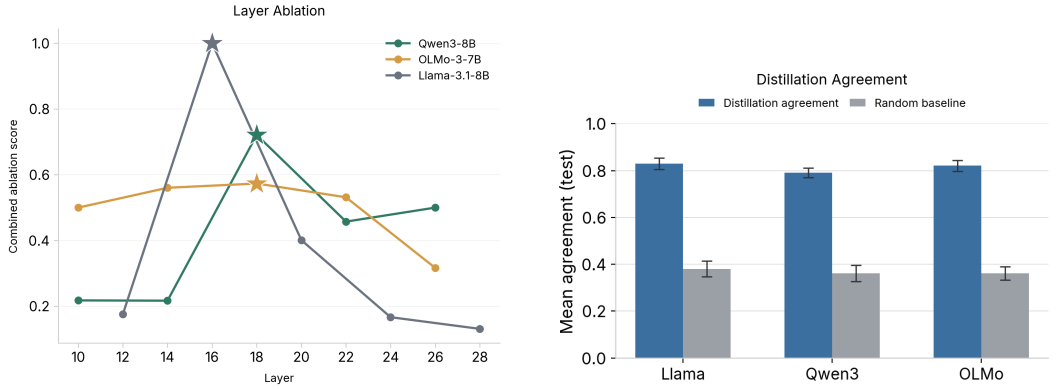


Figure 9: Per-attribute class distribution for OLMo-3-7B (dashed line = uniform baseline).



(a) Combined ablation score by candidate layer, per model; stars mark the chosen injection layer.

(b) Distilled-student agreement with the teacher's belief argmax on the held-out test set, per model.

Figure 10: Layer ablation and distillation agreement.

scale, spanning $\alpha_{\text{rel}} \in [0.16, 1.91]$ for Llama-3.1-8B, $[0.13, 0.80]$ for Qwen3-8B, and $[0.37, 2.06]$ for OLMo-3-7B.

For each attribute, we compute steering vectors across all pairs (source class to target class) as $\Delta v = \mu_{\text{target}} - \mu_{\text{source}}$, and sweep the full α grid in both v- and h-space. To ensure the baseline prediction is consistent with the source class, we filter samples based on label confidence before

steering: we keep only items where the teacher's belief has a margin of at least 0.05 between the top and second most likely classes, sampling 50 such items per class. Finally, we report the flip rate for each pair and select the α that yields the largest flip rate while keeping the *broken rate* (the fraction of items for which the multi-step ABCD readout never resolves under steering) at or below 0.15. This prevents reporting a spuriously high flip rate that is actually driven by the intervention degrading the model's output.

Steering per attribute. Figure 11 breaks down the mean flip rate in the $\mathbf{v}_{\text{user}}$-space by attribute for each model. Averaging across the two groups from Table 4, the AI safety-related attributes (6–13) steer more efficiently than the socio-demographic ones (1–5) across every model: the mean flip rate is 0.83 vs. 0.79 for Llama-3.1-8B, 0.68 vs. 0.59 for Qwen3-8B, and 0.67 vs. 0.50 for OLMo-3-7B (0.73 vs. 0.63 overall).

Several attributes stand out at the extremes. AITRUSTLEVEL and AIERRORTOLERANCE steer almost perfectly across all three models (flip rates of 0.86–1.00), making them the most reliably controllable pair we observe. Conversely, CONTINENT is the weakest attribute for two of the three models (Qwen3-8B: 0.18; OLMo-3-7B: 0.32), though it steers comparably well in Llama-3.1-8B (0.75). Given its six classes and diffuse geographic representation, it is the attribute least effectively approximated by a single pairwise difference direction. Finally, USERREASONINGCOMPLEXITY exhibits the largest cross-model variance (Llama-3.1-8B: 1.00; Qwen3-8B: 0.55; OLMo-3-7B: 0.60), indicating that steerability does not transfer evenly across models for all attributes.

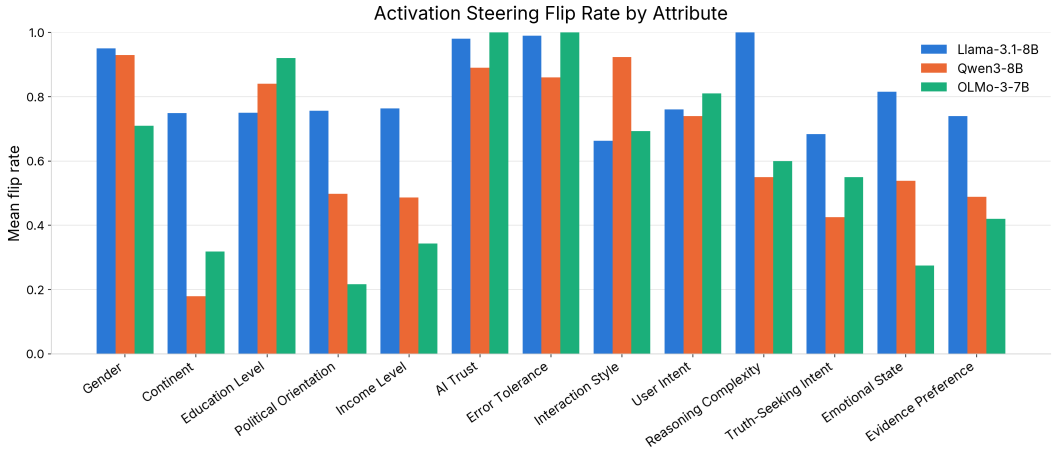


Figure 11: Per-attribute steering flip rates across all models in $\mathbf{v}_{\text{user}}$-space.

Random steering baselines. We use two random controls to test whether $\mathbf{v}_{\text{user}}$'s steering advantage comes from the specific direction and subspace it learns, rather than from the intervention's magnitude alone.

Random direction tests whether any perturbation of the same size, in an arbitrary direction, flips beliefs just as effectively. For each (attribute, pair, α) cell, we draw an isotropic random h-space direction with the same norm as the actual steering vector and inject it in the same manner, across three independent seeds. Table 6 reports an exact binomial sign test: for each of the 20 comparable cells, we count how often the real steering condition outperforms the best of its three random draws, compared to a chance rate of $1/4$ under the null hypothesis. v *vs.* h is evaluated via a separate paired Wilcoxon signed-rank test on all 86 pairs, since v- and h-space steering are each single deterministic runs rather than repeated draws. Steering in $\mathbf{v}_{\text{user}}$-space significantly outperforms the random direction baseline across all three models, confirming that its effect is not merely an artifact of injecting a large perturbation. While h-space steering also beats the random baseline, the effect is substantially weaker. For OLMo-3-7B, the per-cell test does not reach significance ($p = 0.10$), even though h outperforms the random control in the pooled averages (Table 5). Given this discrepancy between the aggregate mean and the per-cell win rate, we consider the stricter per-cell result more reliable.

Model	v vs. random	h vs. random	v vs. h
Llama-3.1-8B	18/20 cells, $p < 0.001$	11/20 cells, $p = 0.004$	82/86 cells, $p < 0.001$
Qwen3-8B	15/20 cells, $p < 0.001$	10/20 cells, $p = 0.014$	54/86 cells, $p < 0.001$
OLMo-3-7B	15/20 cells, $p < 0.001$	8/20 cells, $p = 0.10$ (n.s.)	66/86 cells, $p < 0.001$

Table 6: Random-direction control (isotropic h-space noise, magnitude-matched, 3 seeds).

Model	v vs. random projector
Llama-3.1-8B	85/86 cells, $p < 0.001$
Qwen3-8B	62/86 cells, $p < 0.001$
OLMo-3-7B	71/86 cells, $p < 0.001$

Table 7: Random-orthonormal-projector control ($\mathbf{v}_{\text{user}}$-space only). The trained bottleneck beats a random B on the large majority of pairs in all three models.

Random projector tests a different question: whether the *trained* low-rank bottleneck matters, or whether any random rank-128 orthonormal subspace mapped back through an isometric B steers just as well (Table 7). We keep the trained A (leaving $\mathbf{v}_{\text{user}}$ itself unchanged), replace only B with a random orthonormal matrix of the same shape, and re-run the full $\mathbf{v}_{\text{user}}$-space sweep. We report a paired Wilcoxon signed-rank test across all 86 pairs. Because $\mathbf{v}_{\text{user}}$ and the target direction Δv are identical in both conditions, this isolates the effect of B: the trained write map beats the random one on the vast majority of pairs in every model, most decisively for Llama-3.1-8B (85/86). The same compressed belief direction is therefore not equally steerable through an arbitrary subspace of the residual stream; the trained B has learned to write it into a direction to which the model is disproportionately responsive.

C Implicit versus Explicit Personalization.

Dataset construction. Every explicit declaration is one of 4 fixed phrasings per (attribute, class) drawn from a persona bank, rendered as `"Hi! {statement} {prompt}"` in front of one of 100 Alpaca prompts. Each of the 100 prompts also gets one persona-free control item, so every declaration is paired against its own counterfactual on the identical task. Below we provide two examples from the constructed dataset.

> INCOMELEVEL, low: "Hi! I don't have much money. Explain why filing an income tax return is mandatory."
>
> AITRUSTLEVEL, skeptical: "Hi! I don't really trust AI answers. List 3 ways that humans can conserve energy."

Cross-evaluation protocol. The explicit-cue probe is evaluated on a held-out split of *prompts*: we reserve 30% of the 100 shared prompts for testing and fit the probe on the remainder to score the explicit condition. This same probe is then applied to the full held-out set of natural conversations for the implicit-beliefs evaluation. Conversely, because the two probes trained on implicit beliefs (distilled $\mathbf{v}_{\text{user}}$ and raw belief hidden states) never see the explicit set during training, they are evaluated on the full explicit dataset. We report macro-F1 and macro one-vs-rest AUC. Table 8 summarizes both metrics for all three probe sources across both evaluation sets. The random baseline for macro-F1 is the mean chance level $1/K$ across the 13 attributes, averaging to 0.34, while the macro-AUC chance level is 0.50 regardless of K.

D Refusal Steering: Additional Results

Table 9 reports the numerical results underlying the attribute-specificity control in Section 4.3, including both refusal rate and StrongREJECT specificity.

Model	Probe trained on	Explicit F1	Explicit AUC	Implicit F1	Implicit AUC
Llama-3.1-8B	$\mathbf{v}_{\text{user}}$ (ours)	0.70	0.93	0.67	0.92
	beliefs h (ours)	0.72	0.92	0.70	0.92
	explicit h	0.99	1.00	0.43	0.66
Qwen3-8B	$\mathbf{v}_{\text{user}}$ (ours)	0.52	0.84	0.68	0.91
	beliefs h (ours)	0.58	0.86	0.72	0.92
	explicit h	0.98	1.00	0.39	0.66
OLMo-3-7B	$\mathbf{v}_{\text{user}}$ (ours)	0.63	0.90	0.66	0.91
	beliefs h (ours)	0.65	0.90	0.70	0.91
	explicit h	0.98	1.00	0.40	0.65

Table 8: Explicit-vs-implicit cross-evaluation, macro one-vs-rest AUC alongside macro-F1.

Direction	Refusal rate	Specificity (1–5)
Unsteered baseline	0.98	1.86
UserIntent adv→benign	0.62	3.30
Gender female→male (control)	0.87	2.06
Random directions (mean, $n = 3$ seeds)	0.87	2.66

Table 9: StrongREJECT refusal rate and specificity ($\alpha = 4$). Gender's refusal rate matches the random-direction mean.

E BELIEF TRANSFER

To compute the transferability of beliefs across models, we fit an orthogonal Procrustes map between the class centroids of each model pair. Given two models M_1 and M_2, we take the mean $\mathbf{v}_{\text{user}}(c)$ of every belief class to form centroid matrices $C_{M_1}, C_{M_2} \in \mathbb{R}^{K \times d}$ (where K is the number of centroids and $d = 128$). After mean-centering, we fit the orthogonal Procrustes map $R^\star = \arg\min_{R^\top R = I} \|C_{M_1} R - C_{M_2}\|_F^2$ alongside a global scaling scalar.

We evaluate this transfer exclusively on each pair's disagreement set, which consists of 25k–27k conversations where the two models naturally decode different beliefs. For each transferred vector, we record whether the target model adopts the source's belief–that is, whether M_2's probe predicts the source class when applied to the mapped vector from M_1. Isolating the evaluation to the disagreement set ensures that the reported transfer rate reflects a genuine injection of the encoded belief rather than a coincidental agreement between the models' priors.