PERSONA LEARNING FROM FEEDBACK FOR PERSONALIZED AI ASSISTANTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Personalizing AI assistants requires adapting their behavior to users' preferences and task contexts. Existing work elicits assistant personas through persona prompting or finetuning on persona-conditioned generations. However, persona descriptions alone do not identify what to change or preserve in a particular response, or which behaviors suit the current request. We introduce *Persona Learning from Feedback (PLF)*, a framework that learns reusable, preference-grounded assistant personas as independent LoRA adapters on a shared frozen backbone. During training, a frozen self-teacher conditions on persona specifications and response-specific feedback to provide token-level supervision on each adapter's own generated prefixes. A feedback gate focuses corrective updates on responses judged to need improvement. At deployment, a supervised router uses the request or dialogue to select a single adapter or a fixed composition, without user-specific finetuning. We evaluate PLF on single-turn personalization with model-generated critiques and multi-turn tutoring with simulated students' reactions. Across both settings, PLF improves model-judged preference fit over fixed-persona policies and persona prompting with the same router, demonstrating how feedback-trained behaviors can be reused for adaptive assistant personalization.

1 INTRODUCTION

Personal AI agents are designed to support users across ongoing tasks and goals, using context accumulated through interaction. Effective assistance in these settings depends not only on what users want to accomplish, but also on how they prefer to be helped (Kirk et al., 2024; Sorensen et al., 2024). The same user may want a concise, actionable recommendation when making an urgent decision, but a detailed explanation when exploring an unfamiliar topic. We study this behavioral dimension of personalization: adapting how an assistant helps to the preferences and task requirements expressed in the current request or dialogue.

We represent reusable patterns of assistance through *assistant personas*. Here, a persona specifies preferred response behaviors and behaviors to avoid, such as how much detail to provide or whether to explain a solution directly or guide the user toward it. These personas describe ways of helping that can be reused across users and tasks, rather than fictional identities or permanent user types. We ground their specifications in user preferences rather than a fixed taxonomy of personality traits, and allow different personas to be selected or combined for a request. For example, a user may prefer an answer that is both warm and concise.

Existing work elicits assistant personas through role or style prompting (Xu et al., 2023) and finetuning on persona-conditioned generations (Maiya et al., 2026), with recent approaches also selecting personas according to task utility (Hu et al., 2026). Personalization methods further incorporate retrieved user history (Salemi et al., 2024) or infer response preferences from interaction history (Spohn et al., 2026). These approaches provide ways to specify desired personas or use preference evidence, but a persona description alone does not identify what to change or preserve in a particular response. For example, an instruction to be supportive leaves open whether an answer should acknowledge an overlooked concern or remove repetitive reassurance.

Natural-language feedback can supply these response-specific corrections. Prior work uses critiques to refine outputs (Madaan et al., 2023; Akyürek et al., 2023) or converts textual feedback into training signals through self-distillation (Hübotter et al., 2026). For personalization, however, the desired

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.

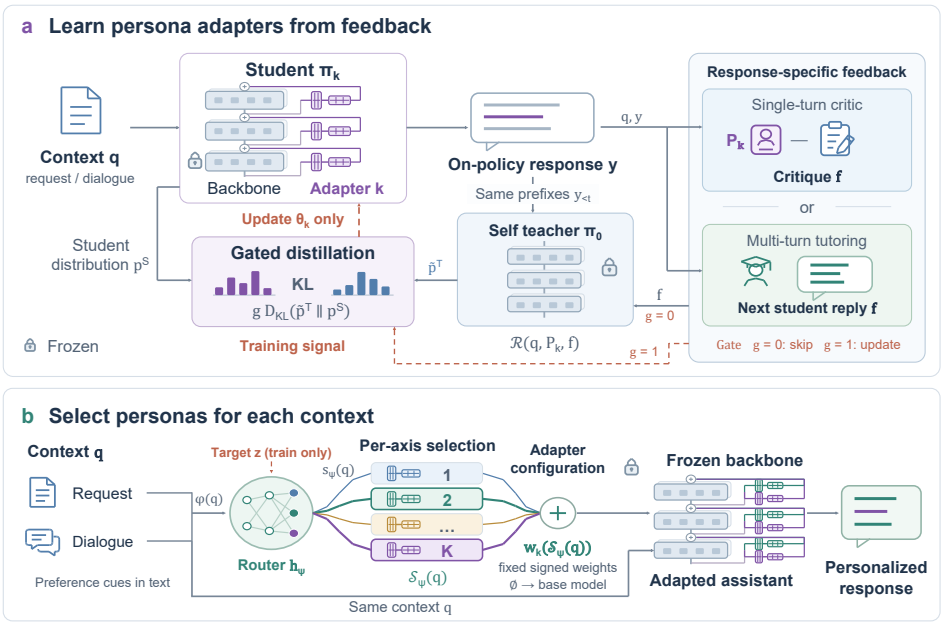

Figure 1: **Persona Learning from Feedback (PLF). (a)** A frozen teacher uses critiques or conversational feedback to supervise persona adapters on student-generated prefixes. A feedback gate excludes satisfactory responses from updates. **(b)** A supervised router selects personas using the request or dialogue, activating a single adapter or a fixed signed composition.

correction is itself preference-dependent: shortening an answer may improve it for one request while making another less useful. This creates two complementary problems: *learning* reusable personas from preference-dependent, response-specific feedback, and *selecting* the appropriate persona or combination from the evidence available before each response.

We introduce *Persona Learning from Feedback (PLF)*, a framework that separates feedback-based persona learning from request-specific persona selection (Figure 1). Each persona is represented by an independent LoRA adapter on a shared frozen backbone (Hu et al., 2022), making its learned parameter updates separately selectable and composable. During training, an adapter generates its own responses and receives feedback evaluated against its intended persona or the corresponding user preference. A feedback gate focuses corrective updates on responses judged to need improvement. A frozen self-teacher conditions on the persona specification and feedback to provide token-level supervision on the student's own generated prefixes (Agarwal et al., 2024; Zhao et al., 2026). The adapter thus learns to express its persona without an additional persona prompt at inference. After persona learning, the adapters are frozen and a supervised router identifies the desired personas from preference evidence expressed in the request or dialogue. The selected personas determine either a single adapter or a signed weight-space composition with validation-selected coefficients. The same learned personas can therefore be reused across users and requests without user-specific finetuning or training a separate adapter for every paired preference.

We evaluate PLF in single-turn personalization and multi-turn tutoring. The single-turn setting uses PRISM requests with assigned preferences grounded in recurring patterns among training users, and model-generated critiques provide response-specific supervision (Kirk et al., 2024). The tutoring setting uses simulated students' subsequent replies as feedback on preceding tutor responses, starting from openings in the Education Dialogue Dataset (Shani et al., 2024). We compare PLF with fixed-persona policies and persona prompting that uses the same router, then examine reuse under combined preferences and unannounced preference changes. Ablations study the effects of feedback gating and the distillation objective. We measure model-judged preference fit separately from simulated student satisfaction and independently judged factual errors in tutoring, distinguishing preferred assistance from stronger persona expression alone. Across both settings, PLF improves average model-judged preference fit over fixed-persona policies and persona prompting with the

same router. The learned personas also support combined preferences and changes revealed through interaction, although composition benefits depend on the personas combined and routing can lag behind a preference change. Our contributions are threefold:

- We formulate assistant personalization as learning reusable, preference-grounded personas from response-specific feedback and selecting them from observable context, without user-specific parameter updates.
- We introduce PLF, which learns independent persona adapters through feedback-gated on-policy self-distillation and reuses them through supervised routing and signed composition.
- We evaluate persona learning and reuse in single-turn personalization and multi-turn tutoring, including combined preferences, unannounced preference changes, and ablations of feedback gating and the distillation objective.

2 Related Work

Assistant personas and user personalization. Persona and role prompting can steer LLM behavior through instructions about role, expertise, style, or interaction strategy, though effects vary across models and tasks (Xu et al., 2023; Salewski et al., 2023; Kong et al., 2024; Zheng et al., 2024). Such behaviors can also be internalized through training. Open Character Training learns persistent character traits from preference data and synthetic dialogues (Maiya et al., 2026), while Hu et al. (2026) identify a trade-off between persona expression and task accuracy and introduce PRISM to selectively activate a distilled persona adapter. Complementary work personalizes behavior using user information. LaMP retrieves from user history (Salemi et al., 2024), Personalized-RLHF learns from heterogeneous user feedback (Li et al., 2025), and APM evaluates preference inference from interaction history while reducing reliance on stereotypical associations (Spohn et al., 2026). Together, these works separate eliciting a behavioral tendency from deciding when it should be used.

Learning from feedback and on-policy distillation. Natural-language feedback can improve model behavior through inference-time critique and revision (Madaan et al., 2023; Akyürek et al., 2023), or through training on feedback-informed revisions and optimization signals derived from critiques (Chen et al., 2024; Wang et al., 2026). On-policy distillation provides teacher supervision on student-generated trajectories, reducing mismatch between training data and the policy's own outputs (Agarwal et al., 2024). Methods condition the teacher on privileged information unavailable to the deployed student (Zhao et al., 2026), including textual feedback in SDPO and RLTF-SD (Hübotter et al., 2026; Song et al., 2026a). Related work applies feedback learning to personalization by distilling information from subsequent user messages (Buening et al., 2026) or using natural-language feedback to improve personalized question answering (Salemi & Zamani, 2026).

Modular personalization and routing. Parameter-efficient and modular adaptation offers an alternative to representing all preferences in a single policy. Personalized Soups trains models for different preference dimensions and combines them through post-hoc parameter merging (Jang et al., 2023), while mixture-of-experts methods select or combine LoRA modules (Wu et al., 2024). For personalized generation, PROPER uses group-level LoRA adaptations with routing over shared and user-specific preferences (Zhang et al., 2025), and CARD combines cluster-level LoRA adaptation with lightweight user-specific personalization (Song et al., 2026b). These approaches motivate modular representations for heterogeneous users and preferences, while leaving open when modularization is preferable to a shared conditional policy.

3 Method

Persona Learning from Feedback (PLF) separates learning how to realize reusable personas in individual responses from selecting the appropriate personas for each request or interaction. Persona specifications define the intended behaviors, response-specific feedback guides their learning, and a supervised router selects or combines the learned behaviors using the observable request or dialogue at deployment (Figure 1). Algorithm 1 summarizes the procedure.

Let q denote the context available before an assistant response, including the current request and any preceding dialogue. PLF assumes a fixed catalog $\mathcal{P} = \{P_k\}_{k=1}^{K}$, where each P_k specifies a reusable assistance behavior associated with a user preference profile. Each persona is represented by an independent LoRA adapter (Hu et al., 2022) with parameters θ_k on a shared frozen base model π_0, yielding policy π_k. Separate adapters keep the learned parameter updates independently selectable and composable. All adapters start with zero weight updates, so $\pi_k = \pi_0$ before training, and each learns its assigned behavior across training contexts. We withhold persona descriptions from the student prompt to train the adapters to express their behaviors without persona prompting. The descriptions instead guide feedback and teacher supervision. At deployment, both the router and the response generator receive q, without a separately supplied preference profile.

3.1 Response-Specific Feedback

A persona specification describes the intended behavior but does not identify how a particular response should change. We obtain this response-specific guidance from feedback evaluated against the persona or its corresponding user preference. For persona k, the current policy samples $y \sim \pi_k(\cdot \mid q)$ and receives natural-language feedback f and an update gate $g \in \{0, 1\}$ (Agarwal et al., 2024).

Single-turn critiques. A critic observes (P_k, q, y) and identifies changes needed to express the intended persona while preserving useful content and satisfying the task. Its critique supplies f. It may instead report that the response needs no revision.

Conversational reactions. In multi-turn tutoring, the adapter interacts with a student simulator conditioned on the learning preference corresponding to P_k. The student's next reply supplies f directly, without a separate textual critique. A separate satisfaction judge reads the tutor response and student reaction to determine whether the student is satisfied. The reaction is unavailable when generating y, but becomes part of the dialogue context for the next response.

Feedback gating. We focus corrective supervision on responses that still need improvement, rather than treating satisfactory responses as further revision targets. We set $g = 0$ for responses judged satisfactory by the critic or satisfaction judge, and $g = 1$ otherwise. Critic outputs with no usable feedback also receive $g = 0$. Both sources yield examples (q, y, f, g) for the same gated learning procedure. Feedback prompts and judgment criteria are specified in Section 4.1.

3.2 Learning Personas from Feedback

Feedback-conditioned teacher. To translate response-level feedback into token-level supervision, we use the frozen base model π_0 as a self-teacher with privileged context. Its input $\mathcal{R}(q, P_k, f)$ augments the original context with the persona specification and feedback as hindsight guidance. The teacher is instructed to answer the original request rather than respond to the feedback. The complete sampled response is not inserted into $\mathcal{R}$ as an earlier draft. For a minibatch indexed by $\mathcal{B}_k$, containing examples (q_i, y_i, f_i, g_i), we score both models on the student's prefixes so that supervision applies to states visited by the current policy:

$$p^{\mathrm{S}}_{i,t} = \pi_k(\cdot \mid q_i, y_{i,<t}), \qquad p^{\mathrm{T}}_{i,t} = \pi_0(\cdot \mid \mathcal{R}(q_i, P_k, f_i), y_{i,<t})\,. \tag{1}$$

The feedback thus informs next-token targets without requiring a separately generated revised answer (Zhao et al., 2026; Hübotter et al., 2026).

Gated distillation. To reduce target storage, we retain and renormalize the teacher's top-M token probabilities to obtain $\widetilde{p}^{\mathrm{T}}_{i,t}$, while keeping the student's full-vocabulary normalization. The adapter minimizes the gated forward-KL objective:

$$\mathcal{L}_{\mathrm{persona}}(\theta_k) = \frac{1}{N_k} \sum_{i \in \mathcal{B}_k} g_i \sum_{t=1}^{|y_i|} D_{\mathrm{KL}}\big(\widetilde{p}^{\mathrm{T}}_{i,t} \,\big\|\, p^{\mathrm{S}}_{i,t}\big)\,, \qquad N_k = \sum_{i \in \mathcal{B}_k} |y_i|. \tag{2}$$

Here $|y_i|$ counts valid response tokens, excluding padding, and N_k includes these tokens even when $g_i = 0$. Examples with $g_i = 0$ contribute no distillation loss. A batch with no positive gates is skipped. Responses, feedback, and teacher targets are held fixed during the update, and gradients reach only θ_k. Feedback collection and distillation alternate using fresh responses from the updated adapters, keeping supervision tied to their current behavior. More details in Appendix A.

3.3 PREFERENCE-BASED ROUTING AND DEPLOYMENT

To match reusable behaviors to request-specific preferences, we freeze the learned adapters and train a router using evidence in the observable context.

Routing supervision. Routing examples are constructed under known preference profiles, which determine their target personas through the predefined preference-to-persona mapping. The router learns to infer these targets from the resulting request or dialogue, without observing the profile itself. Preferences can concern different aspects of assistance, allowing a request to combine behaviors such as a warm tone and a concise explanation. We therefore organize persona indices into J disjoint pairs $\{\mathcal{I}_a\}_{a=1}^{J}$, each containing two contrasting alternatives on a preference axis, so $K = 2J$. The routing dataset $\mathcal{D}_{\text{route}} = \{(q_i, \mathbf{z}_i)\}_{i=1}^{n}$ contains nonzero targets $\mathbf{z}_i \in \{0,1\}^K$ satisfying $\sum_{k\in\mathcal{I}_a} z_{i,k} \le 1$ for each axis.

Router training and selection. A lightweight classifier h_ψ receives representations $\varphi(q)$ from the frozen base model with adapters disabled. We train it to predict the persona targets:

$$\mathcal{L}_{\text{route}}(\psi) = \frac{1}{n}\sum_{i=1}^{n} \ell_{\text{CE}}(h_\psi(\varphi(q_i)), \mathbf{z}_i)\,. \tag{3}$$

Here ℓ_{CE} is mean binary cross-entropy over personas for multi-label routing or categorical cross-entropy for exclusively single-label routing. Only ψ is trained. Let $\mathbf{s}_\psi(q)$ denote the corresponding sigmoid or softmax scores. To avoid selecting contrasting behaviors together, we retain at most one persona per axis, choosing its highest-scoring alternative when the score exceeds a fixed threshold:

$$\begin{aligned} \widehat{k}_a(q) &= \arg\max_{k\in\mathcal{I}_a} s_{\psi,k}(q), \\ \mathcal{S}_\psi(q) &= \left\{ \widehat{k}_a(q) : a = 1,\dots,J,\ s_{\psi,\widehat{k}_a(q)}(q) > \tau \right\}. \end{aligned} \tag{4}$$

For mandatory selection, the threshold condition is omitted.

Adapter configuration. The selected personas determine fixed adapter coefficients, while router scores are used only for selection. At an adapted layer ℓ, let W_ℓ^0 be the base weight matrix and $\Delta W_{\ell,k}$ persona k's LoRA update. Generation uses

$$W_\ell(q) = W_\ell^0 + \sum_{k=1}^{K} w_k(\mathcal{S}_\psi(q))\,\Delta W_{\ell,k}. \tag{5}$$

A single selection activates only its corresponding adapter with coefficient one, while an empty selection uses the base model as a fallback. To support combined preferences without training a separate adapter for every pair, we compose the learned updates in weight space. Given a selected pair $\mathcal{S} = \{k_a, k_b\}$ from distinct axes, let $\bar{k}_c$ denote the opposite persona to k_c on axis c. We use the coefficients

$$w_{k_c}(\mathcal{S}) = \frac{1}{4} + \frac{\lambda_c}{2}, \qquad w_{\bar{k}_c}(\mathcal{S}) = \frac{1}{4} - \frac{\lambda_c}{2}, \qquad c \in \{a, b\}, \tag{6}$$

with all other coefficients zero. This combines the mean of the four participating updates with scaled within-axis parameter differences. The strengths λ_c are selected on validation data and fixed at evaluation, rather than predicted by the router. The signed sum is represented exactly as one adapter through the LoRA factor concatenation described in Appendix A. The adapter configuration remains fixed during a response and is recomputed from the updated context before each subsequent response. Section 4.1 specifies the persona catalog, routing-data construction, classifier and selection configurations, and composition-strength settings.

4 EXPERIMENTS

We evaluate Persona Learning from Feedback (PLF) in single-turn personalization and multi-turn tutoring. We ask whether natural language feedback produces useful persona heads, whether those heads support combined or changing preferences, and how the training design affects their behavior.

Algorithm 1 Persona Learning from Feedback (PLF)

Require: Frozen base π_0, personas $\{P_k\}$, context and feedback sources, and routing data $\mathcal{D}_{\text{route}}$.

Persona learning

Initialize adapters with zero weight updates.
for each training cycle and each persona k **do**
  Collect $\{(q_i, y_i, f_i, g_i)\}_{i \in \mathcal{B}_k}$ on-policy from π_k.
  $N_k \leftarrow \sum_{i \in \mathcal{B}_k} |y_i|$.
  if $\sum_{i \in \mathcal{B}_k} g_i > 0$ **then**
    Compute teacher targets $\widetilde{p}^{\mathrm{T}}_{i,t}$ on $y_{i,<t}$ for $g_i = 1$.
    Update only θ_k using Eq. 2.
  end if
end for

Router training

Freeze adapters and compute $\varphi(q_i)$ with π_0 on $\mathcal{D}_{\text{route}}$.
Train ψ using Eq. 3.

Deployment for each context q

Score $\mathbf{s}_\psi(q)$ and select $\mathcal{S}_\psi(q)$ using Eq. 4.
Generate from q with fixed weights $w_k(\mathcal{S}_\psi(q))$ using Eq. 5.

4.1 Experimental Setup

Single-turn setting: preferences in a request. We use opening requests from PRISM (Kirk et al., 2024). Following Kirk et al. (2026), recurring survey preferences among training users motivate two persona axes: *warm* versus *impersonal* voice, and *concise* versus *thorough* depth. They capture common differences in desired tone and detail, while permitting combinations such as warm and concise. We split requests by user and remove exact and near-duplicate tasks across splits, yielding 6,400 training, 203 validation, and 500 test requests. For routing and evaluation, we minimally rewrite each request to express one of four individual personas or four cross-axis pairs (Spohn et al., 2026). The assigned preference, rather than the user's original survey response, is the evaluation target. During head training, a Qwen3.5-27B critic reads the original request, the head's answer, and its persona card, then explains what to change and preserve.

Multi-turn setting: preferences revealed through interaction. We use teacher openings from the Education Dialogue Dataset (Shani et al., 2024): 4,524 training, 200 validation, and 500 test dialogues, with topics held out across splits. All subsequent turns are generated. Our personas are *direct instruction*, which explains and demonstrates, and *interactive guidance*, which asks questions and lets the student reason. A Qwen3.5-27B student simulator holds a hidden preference and reacts to each tutor turn. The following student's turn is the feedback, rather than a separate critique.

Training. Both settings freeze the backbone and train one LoRA head per persona, using Qwen3-4B-Instruct-2507 (Yang et al., 2025) for single-turn responses and Llama-3.1-8B-Instruct (Grattafiori et al., 2024) for tutoring. The frozen base, conditioned on the persona card and feedback, teaches the head on its own sampled response prefixes through forward KL distillation. A no-change feedback gate excludes satisfactory responses from the loss. We select single-turn heads on validation and use the final heads from the tutoring run. After freezing the heads, we train MLP routers on the backbone's last-token representation: the single-turn router selects persona indicators from the request, while the tutoring router selects a head after each student reply. Data, training settings, and prompts details can be found in Appendices B–D.

Evaluation. In both settings, we compare routed PLF against the frozen base model, each learned persona head used as a fixed policy, and router-based persona prompting (Spohn et al., 2026), which uses the same router as PLF to select persona descriptions for the base model's prompt. For *single-turn composition*, we additionally compare with prompting using both requested personas. On validation, we also compare three alternative mixing rules: summing adapter updates, averaging adapter updates, and mixing head logits. For *multi-turn tutoring*, we include an oracle that selects the head matching the student's hidden preference, with anticipating and one-turn-lagged variants for preference switching. Supplementary tutoring controls include oracle persona prompting, linear routers, and the base without a persona prompt. Our primary metric in both settings is *pairwise preference win rate*: a separate `gpt-6-luna` judge evaluates which response or complete dialogue better fits the assigned preference. Each pair is judged in both presentation orders; disagreements count as

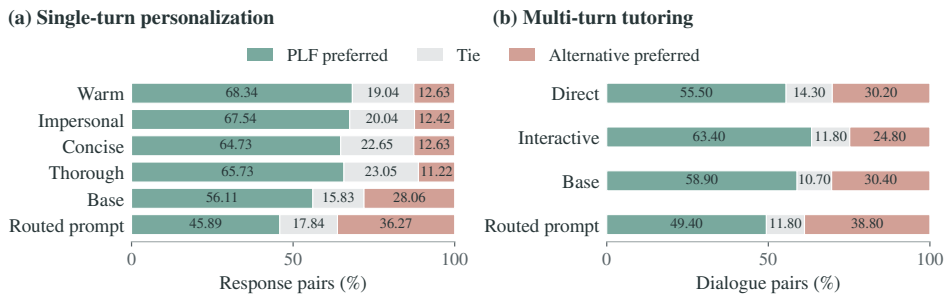

Figure 2: **Preference fit of feedback-trained personas.** Percentages of PLF wins, ties, and alternative wins on (a) 499 single-turn requests and (b) 500 multi-turn tutoring dialogues. Routed prompt uses the same router as PLF.

Table 1: **Single-turn composition.** Mixture win rates (%) against fixed heads.

Pair	Warm	Impers.	Conc.	Thorough
Warm + concise	**93**	94	**92**	99
Warm + thorough	**88**	94	98	**68**
Impers. + concise	80	**60**	**48**	81
Impers. + thorough	89	**75**	85	**64**

Table 2: **Multi-turn preference switching.** PLF win rates (%) against alternatives.

Switch	Direct	Inter.	Base	Oracle
Direct → interactive	**55**	81	54	47
Interactive → direct	67	**56**	59	40
Mean	61	68	56	43

Left: 222/247 paired requests routed correctly, with bold marking constituent heads. Right: 500 openings per direction, with bold marking the initially matching head. Full results appear in Appendix E.

ties, and ties receive half credit. For tutoring, we also report *student satisfaction*, the share of student replies without flagged objections, and *factual error rate*, the share of tutor turns flagged by a separate fact-checking judge. Evaluation protocols, decoding settings, and full comparisons appear in Appendices B.3 and E.3.

4.2 Can Natural Language Feedback Learn Useful Personas?

Learned heads improve on both fixed personas and prompting. In single-turn personalization, PLF is preferred over every fixed head and the base model (Figure 2a). No one persona serves the heterogeneous test preferences as well as selecting and combining behaviors. PLF also achieves a 55% win rate against persona prompting with the same router. This comparison holds preference selection fixed, showing that learned heads add value beyond identifying a suitable persona description. The smaller advantage over routed prompting than over fixed heads is consistent with preference matching contributing substantially to the full system's gain.

Student reactions provide useful feedback without an explicit critique. Tutoring shows a similar advantage over fixed heads, the base, and routed prompting (Figure 2b). Thus, feedback embedded in an ongoing interaction can also support persona learning. PLF approaches the oracle on average, although a correctly matched fixed head can still outperform it, especially for direct-instruction students (Appendix E.3). This identifies routing as a remaining source of error: useful heads do not ensure that the right one is always selected. Compared with routed prompting, PLF also improves student satisfaction and reduces flagged factual errors, suggesting that its preference advantage is not achieved by sacrificing measured tutoring quality.

4.3 Can Learned Personas Serve Combined or Changing Preferences?

Single-turn experiment: combine voice and depth. We test composition on requests expressing both a voice and a depth preference. Heads are trained only on individual personas; the router sees both single and paired labels during training. At inference, we combine adapter weights using a validation-selected rule that adjusts voice and depth strength separately (Appendix E.4). Table 1 isolates composition on correctly routed requests.

Composition helps most when the behaviors are complementary. Warm-concise gives the clearest improvement over both constituent heads: the mixture retains a short answer while adding en-

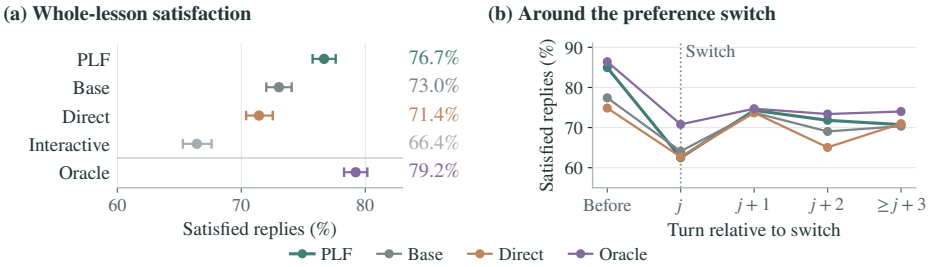

Figure 3: **Student satisfaction under changing preferences.** Share of replies without flagged objections, averaged across switch directions: (a) whole lessons, with 95% bootstrap intervals; (b) aligned to the switch at j. Only the oracle knows the new preference when choosing turn j.

Table 3: **Examples of composition and adaptation.** Saved responses from a paired request and two tutoring switches.

User request or feedback	Model response
Warm + concise (single turn) "how do i get better at playing guitar? i'd love a quick, warm tip!"	*Concise head:* "Practice every day, even just 10 minutes." *PLF mixture:* "Play every day, even just 10 minutes. **Consistency beats perfection.**"
Direct → interactive (tutoring) "You're just telling me facts again; **ask me a real question instead.**"	*Next turn, interactive head:* "Let's put this into practice: if a car has a mass of 1500 kilograms and is subjected to a force of 3000 Newtons, **what is its acceleration?**"
Interactive → direct (tutoring) "**Just tell me the answer** instead of making me guess what you already know."	*Next turn, direct head:* "Time and a thief both **take something valuable away without asking permission.**"

couragement. In contrast, impersonal-concise offers little advantage over the concise head, whose terse style already overlaps with impersonality. Including routing errors preserves this pattern but reduces the warm–concise gains. Composition is also limited by head quality: thorough mixtures improve on their constituent heads yet remain less preferred than prompting with both persona cards. The combination rule matters as well. On validation, adjusting the two axes outperforms simply summing or averaging the desired heads; thorough pairs require weaker voice weights to preserve depth. Together, these results support reusing learned behaviors, while showing that their relative strengths need calibration.

Multi-turn experiment: respond to a preference switch. For each test opening, the student's preference switches once, in both directions, at a uniformly sampled turn $j \in \{2, \ldots, 7\}$. The switch is unannounced: its first evidence is the student's reply to turn j, so the router can first respond at $j + 1$. The router is frozen and was trained only on fixed-preference dialogues. We compare it with fixed heads, the base, and an oracle that knows the new preference in advance and switches at turn j. The judge sees both preferences and the marked switch point.

Routing transfers to changing preferences, but adaptation lags. Although its router is trained only on fixed-preference dialogues, PLF is preferred over each fixed head and the base in both switch directions (Table 2). It also achieves the highest whole-lesson satisfaction among the evaluated non-oracle policies, about 2.5% below the anticipating oracle (Figure 3a). Selecting personas from the evolving dialogue therefore offers an advantage over committing to either teaching style throughout the lesson. Figure 3b shows a marked satisfaction rebound at $j+1$, the first turn that can use feedback reflecting the new preference. Head selection still lags: PLF switches at its first opportunity in about 40% of dialogues, with greater difficulty after shifts toward direct instruction. Recovery remains partial, and PLF does not lead at every later turn; its advantage is clearest in preference fit and satisfaction over complete dialogues. Detailed delay diagnostics appear in Appendix E.5.

Qualitative examples. Table 3 shows how these behaviors appear in responses. For the guitar request, the mixture adds encouragement to a brief practice tip and is preferred to both constituent heads in both judge orders. The tutoring examples show who supplies the reasoning: the tutor poses a problem when interactive and explains directly when prompting is disabled. These selected examples show successful switches, while the aggregate results capture adaptation delays.

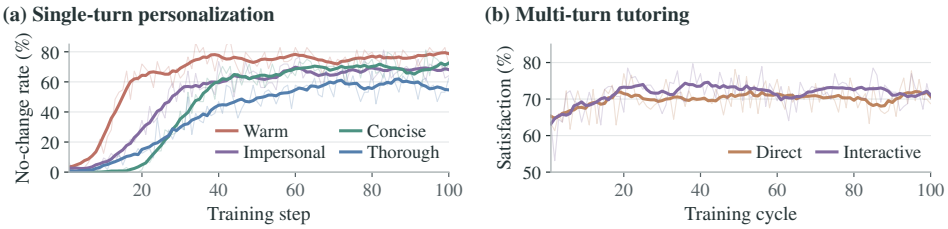

Figure 4: **Feedback gating during persona learning.** (a) Critic no-change rate in single-turn personalization. (b) Satisfaction-judge acceptance in multi-turn tutoring before additional gating checks. Higher values indicate more responses passing the displayed feedback check. Thin lines are raw values and thick lines are centered nine-point means.

4.4 Ablations: Feedback Gating and KL Direction

We examine whether focusing corrective updates on responses that need improvement helps persona learning, and how the direction of the distillation objective affects preference fit.

Learning progress differs across personas. During gated training, the warm and impersonal adapters reach high critic approval earlier than the concise adapter, while the thorough adapter improves more slowly and finishes with the lowest approval (Figure 4a). This pattern is consistent with the weaker thorough compositions in RQ2. In tutoring, training satisfaction improves mostly early and then plateaus (Figure 4b). These curves measure training feedback rather than held-out performance, and the tutoring curve reports satisfaction-judge acceptance before the additional checks used by the final gate.

Feedback gating improves preference fit selectively. At matched checkpoints, the gated single-turn system achieves a 56% win rate over ungated training, while the gated interactive tutoring adapter achieves 55% and the direct adapters remain near parity. Judge rationales associate the warm-persona gain with avoiding reassurance that displaces requested information, whereas thorough-only requests favor ungated training. The gated interactive persona also asks questions more often and receives fewer repetition complaints, with a held-out preference advantage remaining after its early training advantage fades. Critic approval is nevertheless similar with and without gating, suggesting that approval alone does not explain the gain. Updates on already approved responses may reinforce an expressed persona, while slower-learning personas may still benefit from continued supervision. Moreover, because N_k in Eq. 2 includes tokens from gated-out examples, gating changes both which responses contribute and the effective loss scale. The ablation therefore does not isolate these effects. Full breakdowns and controls appear in Appendix G.

KL direction has persona-dependent effects. Forward and reverse KL are near parity overall at training step 60, with a 51% forward-KL win rate across 499 disjointly assigned requests. When every request is evaluated under each persona, forward KL is preferred for impersonal responses and reverse KL for thorough responses, while warm is inconclusive and the concise advantage does not persist. Under reconstructed teacher contexts, reverse-KL adapters have lower entropy, closer to the teacher's, while top-token agreement changes little. Forward-KL impersonal responses use more hedging, whereas reverse-KL thorough responses are longer, consistent with the corresponding judge preferences. These results suggest that probability allocation beyond the top token can affect persona fit, although the diagnostics do not separate this effect from sparse teacher support and target approximation. Full KL analyses appear in Appendix H.

5 Conclusion

We introduced Persona Learning from Feedback (PLF), which learns reusable persona adapters from response-specific feedback through on-policy self-distillation and selects or combines them using contextual preference evidence. Across single-turn personalization and simulated multi-turn tutoring, PLF improves model-judged preference fit over fixed persona policies, the base model, and persona prompting with the same router. The learned personas also support combined and changing preferences. These findings support separating persona learning from contextual selection as an approach to personalization without user-specific parameter updates.

AI USE STATEMENT

In this work, we used generative AI tools to assist with language editing, including improving grammar, readability, and clarity of the manuscript. We have not used generative AI tools for generating research ideas, conducting analyses, interpreting results, or formulating scientific conclusions, and other disclosure-related uses are not applicable to this work. Additionally, generative AI tools were used to refine the presentation and wording of selected passages without altering their substantive meaning. All AI-assisted text was carefully reviewed and edited by the authors to ensure that it accurately reflects our original ideas, findings, and interpretations. We take full responsibility for the final content of this work, including all text, claims, conclusions, and any material produced with the aid of generative AI.

ETHICS STATEMENT

This work uses existing research datasets, model-generated feedback, simulated tutoring interactions, and a small human evaluation. The human study was approved by the relevant Institutional Review Board (IRB) before data collection, and all participants provided informed consent. Our persona categories are derived from response-style preferences rather than demographic or sensitive attributes. The tutoring experiments use simulated students, so their satisfaction measures should not be interpreted as evidence of human learning outcomes. More broadly, personalization systems may incorrectly infer user preferences or reduce them to coarse categories, and we do not intend PLF for high-stakes profiling or decision making.

REPRODUCIBILITY STATEMENT

We provide the algorithm, model and training configurations, data processing and split procedures, routing and decoding settings, evaluation protocols, prompts, and statistical procedures needed to reproduce our experiments. Appendices A–D give implementation, data, training, routing, and prompt details, while Appendices E–H provide the full evaluation results and ablation protocols. We also report model versions, random seeds, hardware settings, and dataset revisions where applicable.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Afra Feyza Akyürek, Ekin Akyürek, Ashwin Kalyan, Peter Clark, Derry Tanti Wijaya, and Niket Tandon. Rl4f: Generating natural language feedback with reinforcement learning for repairing model outputs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7716–7733, 2023.

Thomas Kleine Buening, Jonas Hübotter, Barna Pásztor, Idan Shenfeld, Giorgia Ramponi, and Andreas Krause. Aligning language models from user interactions. *arXiv preprint arXiv:2603.12273*, 2026.

Angelica Chen, Jérémy Scheurer, Jon Ander Campos, Tomasz Korbak, Jun Shern Chan, Samuel R Bowman, Kyunghyun Cho, and Ethan Perez. Learning from natural language feedback. *Transactions on machine learning research*, 2024.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Zizhao Hu, Mohammad Rostami, and Jesse Thomason. Expert personas improve llm alignment but damage accuracy: Bootstrapping intent-based persona routing with prism. *arXiv preprint arXiv:2603.18507*, 2026.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, et al. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026.

Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong Wang, Jack Hessel, Luke Zettlemoyer, Hannaneh Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu. Personalized soups: Personalized large language model alignment via post-hoc parameter merging. *arXiv preprint arXiv:2310.11564*, 2023.

Hannah Rose Kirk, Alexander Whitefield, Paul Röttger, Andrew Michael Bean, Katerina Margatina, Rafael Mosquera, Juan Manuel Ciro, Max Bartolo, Adina Williams, He He, Bertie Vidgen, and Scott A. Hale. The PRISM alignment dataset: What participatory, representative and individualised human feedback reveals about the subjective and multicultural alignment of large language models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=DFr5hteojx.

Hannah Rose Kirk, Liu Leqi, Fanzhi Zeng, Henry Davidson, Bertie Vidgen, Christopher Summerfield, and Scott A Hale. Prism-x: Experiments on personalised fine-tuning with human and simulated users. *arXiv preprint arXiv:2605.13307*, 2026.

Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, Xin Zhou, Enzhi Wang, and Xiaohang Dong. Better zero-shot reasoning with role-play prompting. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4099–4113, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.228. URL https://aclanthology.org/2024.naacl-long.228/.

Xinyu Li, Ruiyang Zhou, Zachary Chase Lipton, and Liu Leqi. Personalized language modeling from personalized human feedback, 2025. URL https://openreview.net/forum?id=bqUsdBeRjQ.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=S37hOerQLB.

Sharan Maiya, Henning Bartsch, Nathan Lambert, and Evan J Hubinger. Open character training: Shaping the persona of AI assistants through constitutional AI, 2026. URL https://openreview.net/forum?id=X9MMGZdqmc.

Alireza Salemi and Hamed Zamani. Learning from natural language feedback for personalized question answering. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1554–1565, 2026.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. Lamp: When large language models meet personalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7370–7392, 2024.

Leonard Salewski, Stephan Alaniz, Isabel Rio-Torto, Eric Schulz, and Zeynep Akata. In-context impersonation reveals large language models' strengths and biases. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=CbsJ53LdKc.

Lior Shani, Aviv Rosenberg, Asaf Cassel, Oran Lang, Daniele Calandriello, Avital Zipori, Hila Noga, Orgad Keller, Bilal Piot, Idan Szpektor, Avinatan Hassidim, Yossi Matias, and Remi Munos. Multi-turn reinforcement learning with preference human feedback. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=rVSc3HIZS4.

Yuda Song, Lili Chen, Fahim Tajwar, Remi Munos, Deepak Pathak, J Andrew Bagnell, Aarti Singh, and Andrea Zanette. Expanding the capabilities of reinforcement learning via text feedback. *arXiv preprint arXiv:2602.02482*, 2026a.

Yutong Song, Jiang Wu, Weijia Zhang, Chengze Shen, Shaofan Yuan, Weitao Lu, Jian Wang, Yu Wang, Nikil Dutt, and Amir M Rahmani. Card: Cluster-level adaptation with reward-guided decoding for personalized text generation. *arXiv preprint arXiv:2601.06352*, 2026b.

Taylor Sorensen, Jared Moore, Jillian Fisher, Mitchell Gordon, Niloofar Mireshghallah, Christopher Michael Rytting, Andre Ye, Liwei Jiang, Ximing Lu, Nouha Dziri, et al. A roadmap to pluralistic alignment. *arXiv preprint arXiv:2402.05070*, 2024.

Philipp Spohn, Leander Girrbach, and Zeynep Akata. Apm: Evaluating style personalization in llms with arbitrary preference mappings. *arXiv preprint arXiv:2605.21063*, 2026.

Hanyang Wang, Lu Wang, Chaoyun Zhang, Tianjun Mao, Si Qin, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. Text2grad: Reinforcement learning from natural language feedback. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=SIE9fNq8lk.

Xun Wu, Shaohan Huang, and Furu Wei. Mixture of loRA experts. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=uWvKBCYh4S.

Benfeng Xu, An Yang, Junyang Lin, Quan Wang, Chang Zhou, Yongdong Zhang, and Zhendong Mao. Expertprompting: Instructing large language models to be distinguished experts. *arXiv preprint arXiv:2305.14688*, 2023.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Linhai Zhang, Jialong Wu, Deyu Zhou, and Yulan He. PROPER: A progressive learning framework for personalized large language models with group-level adaptation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16399–16411, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.800. URL https://aclanthology.org/2025.acl-long.800/.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

Mingqian Zheng, Jiaxin Pei, Lajanugen Logeswaran, Moontae Lee, and David Jurgens. When "a helpful assistant" is not really helpful: Personas in system prompts do not improve performances of large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 15126–15154, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.888. URL https://aclanthology.org/2024.findings-emnlp.888/.

A PLF Implementation Details

A.1 Truncated Teacher Targets

Teacher conditioning is defined in Section 3.2.

For persona k's batch, let $\mathcal{V}_{i,t}^{(M)}$ contain the indices of the M most probable tokens under $p_{i,t}^{\mathrm{T}}$. The stored target is

$$\widetilde{p}_{i,t}^{\mathrm{T}}(v) = \frac{\mathbf{1}\{v \in \mathcal{V}_{i,t}^{(M)}\} p_{i,t}^{\mathrm{T}}(v)}{\sum_{u \in \mathcal{V}_{i,t}^{(M)}} p_{i,t}^{\mathrm{T}}(u)}. \tag{7}$$

We use $M = 256$ to reduce target storage and transfer costs. The student retains full-vocabulary normalization and is not renormalized over the teacher's retained support. The teacher target is fixed during each update. In Eq. 2, $|y_i|$ excludes padding, and N_k counts valid tokens even when $g_i = 0$. The loss is not renormalized over only the examples selected by the gate. A gated-out example has no distillation contribution; this does not imply that updates from other examples leave its future response distribution unchanged.

A.2 LoRA Parameterization and Exact Composition

For each adapted weight matrix, persona k contributes

$$\Delta W_{\ell,k} = \frac{\alpha_{\mathrm{L}}}{r_{\mathrm{L}}} B_{\ell,k} A_{\ell,k}, \tag{8}$$

where $A_{\ell,k} \in \mathbb{R}^{r_{\mathrm{L}} \times d_{\mathrm{in}}}$ and $B_{\ell,k} \in \mathbb{R}^{d_{\mathrm{out}} \times r_{\mathrm{L}}}$. We initialize $B_{\ell,k} = 0$, so each persona policy initially recovers the base model. Persona adapters have disjoint parameters, and the shared base weights remain frozen.

Let $k_1, \ldots, k_m$ index the adapters with nonzero composition coefficients, and suppress the dependence of the coefficients on $\mathcal{S}$. Exact weight-space composition uses

$$A_\ell^{\mathrm{mix}} = \begin{bmatrix} A_{\ell,k_1} \\ \vdots \\ A_{\ell,k_m} \end{bmatrix}, \qquad B_\ell^{\mathrm{mix}} = \begin{bmatrix} w_{k_1} B_{\ell,k_1} & \cdots & w_{k_m} B_{\ell,k_m} \end{bmatrix}. \tag{9}$$

Keeping the same ratio of LoRA scale to rank gives

$$\frac{\alpha_{\mathrm{L}}}{r_{\mathrm{L}}} B_\ell^{\mathrm{mix}} A_\ell^{\mathrm{mix}} = \sum_{j=1}^{m} w_{k_j} \Delta W_{\ell,k_j}. \tag{10}$$

The paired implementation on Qwen3-4B uses four adapters with $r_{\mathrm{L}} = 32$ and $\alpha_{\mathrm{L}} = 64$ per adapter. Concatenation produces a rank-128 parameterization with scale 256, preserving $\alpha_{\mathrm{L}}/r_{\mathrm{L}} = 2$. The result is one composite adapter and does not require generating responses from separate policies.

A.3 Signed Pair Composition

For the selected pair $\mathcal{S} = \{k_a, k_b\}$, the coefficients in Eq. 6 give

$$\begin{aligned} \Delta W_\ell^{\mathrm{pair}} = \sum_{c \in \{a,b\}} \Big[& \frac{1}{4} \left(\Delta W_{\ell,k_c} + \Delta W_{\ell,\bar{k}_c} \right) \\ & + \frac{\lambda_c}{2} \left(\Delta W_{\ell,k_c} - \Delta W_{\ell,\bar{k}_c} \right) \Big]. \end{aligned} \tag{11}$$

This is the mean of the four participating updates plus one weighted contrast per selected axis. The coefficients sum to one but need not be nonnegative. The decomposition is exact in parameter space; it does not imply that the contrasts have independent behavioral effects.

B Additional Experiment Details

This section describes the data, training configurations, routing procedures, and evaluation protocol used in Section 4. Each persona head is an independent LoRA adapter on a frozen backbone. Both settings use rank 32, LoRA scale $\alpha_{\mathrm{L}} = 64$, no adapter dropout, and adapters on all attention and MLP projections. Heads sample at temperature 0.7 and are trained with AdamW at learning rate 10^{-5} and 3% warmup. The frozen backbone supplies feedback-conditioned teacher targets, retaining its top-256 tokens as specified in Appendix A.

B.1 Single-Turn Data, Training, and Routing

Data source and preference screening. We use the opening requests and survey preferences from PRISM (Kirk et al., 2024), joining `conversations.opening_prompt` and `survey.system_string` by `user_id`. Only the first user message is retained from each conversation. The source is `HannahRoseKirk/prism-alignment` at revision `18ab5cfb37456f4ec8cbc00212ce54cf7b1239f6`. It contains 8,011 conversations from 1,396 users who entered the conversation phase.

Preferences are manually screened for whether a single response from a model without tools can satisfy them. The screening uses only the preference text, without requests or demographics, and includes a blind reread of borderline cases and a random 10% of preferences. We exclude 198 users whose preferences request sourcing or verification and 31 whose preferences primarily concern unavailable capabilities, such as cross-conversation memory, browsing, or real-world actions. Mixed preferences requesting sources are excluded in full rather than removing selected clauses. This avoids evaluating unsupported sourcing requirements in a setting without tools. We retain 1,167 users, including 96 whose otherwise actionable preferences contain a minor capability-related clause. Requests for honesty or acknowledgment of uncertainty are retained.

Persona catalog. Following the use of group-level preferences in Kirk et al. (2026), we code the retained preferences for response style, excluding topics, capability requests, and shared requirements such as accuracy, honesty, safety, and respect. Co-occurring style codes define warm and impersonal voice groups and concise and thorough depth groups. A user is assigned by the style they emphasize most, or by the first secondary style associated with a group when the primary style has no group. Users with no grouped style enter a base group, while a separate clarifying-question group is too small to train as a persona.

Group summaries use only training users, with 254 warm, 326 impersonal, 127 concise, and 75 thorough users. Each summary retains response-style requirements expressed by at least 20% of the group's training users and omits clauses that cannot be judged from the response without knowing the correct content. The resulting persona cards each describe one axis and are fixed before the reported training and evaluation. Their exact wording is given in Appendix C.1. The original survey preferences and group summaries are not supplied as separate conditioning text to the student, teacher, critic, or router. The teacher and critic receive the target persona card instead.

Splits and request cleaning. Splits are stratified by group and keep all requests from each user together, targeting approximately 100 test requests and 40 validation requests per evaluation group. User selection uses the fixed salt `prism-v2-stratify-v1`. The selection first considers users held out in earlier runs, then previously uninspected training users in salted-hash order. The 239 users whose data had been inspected during design are excluded from validation and test. Exact duplicate requests and contentless openings, such as greetings without a task or the literal `EMPTY STRING`, are removed. Near-duplicate requests are identified when token-set Jaccard similarity is at least 0.90, or when it is at least 0.75 with containment at least 0.95 and the shorter request contains at least five tokens. Candidate matches are blocked using tokens with document frequency at most 100 and grouped into connected components. When a component spans splits, lower-priority rows are excluded without moving requests between splits. The final splits share no users or exact or near-duplicate requests under these rules. Table 4 distinguishes the original-request splits from their usable rewritten versions.

The head-training file receives an additional manual review of every request with at most four words and every request containing profanity or sexual terms, together with a scan of longer requests for gibberish or insults without a task. This review removes 31 requests without a usable task

Table 4: **Single-turn data splits.** User counts refer to the original-request splits after filtering and deduplication. Rewritten requests are used for router training or evaluation, not for persona-head training. The final head-training corpus is separately expanded to 6,400 original requests from 1,242 users.

Split	Users	Original requests	Usable rewrites
Train	1,033	5,698	5,677
Validation	39	203	203
Test	89	500	499

and replaces them with additional requests. It does not remove requests solely because they are short, controversial, in another language, or concern sexual topics. The final corpus contains 5,671 requests from the original training split and 729 additional opening requests from users excluded by the preference filter. These additional users can supply tasks because head training does not use their survey preferences. Candidate additions are checked against all splits under the same duplicate rules. Sampling uses seed 20260925, with seed 20260926 for the 31 replacements. The resulting 6,400 requests from 1,242 users support one pass of 100 batches with 64 requests each.

Preference-embedded requests. Following the prompt-rewriting approach of Spohn et al. (2026), router training and evaluation use minimally rewritten requests that express an assigned preference through the user's message. Each original request receives one of eight conditions: four individual personas or four cross-axis pairs. Conditions are balanced before rewriting using a seeded shuffle with `crc32('pref-assign|<split>')`. The assigned condition, rather than the original user's survey response, supplies the router target and the preference judge's reference. The four condition labels and the rewriting prompt are given in Appendix C.4.

Rewriting uses Qwen3.5-27B with thinking disabled, temperature 1.0, and at most 400 generated tokens. The seed is `crc32('pref-rw|<split>|<example_id>')` plus the attempt index, with up to two retries after a refusal. After retries, 21 training requests and one test request have no usable rewrite, leaving 5,677 training, 203 validation, and 499 test requests. The test set contains 247 paired-preference requests.

A reader check predicts the voice and depth preferences from the rewritten message, and a same-task check compares the original and rewritten requests. Both checks use Qwen3.5-27B at temperature zero, and their templates appear in Appendix C.5. Their outputs are recorded but do not determine which examples are retained. Consequently, the rewritten sets are not restricted to examples that pass these checks. Warm-only rewrites often convey warmth through a greeting rather than an explicit preference, and impersonal wording can also suggest a preference for concision. The benchmark therefore evaluates responses against the assigned condition without assuming that every rewrite expresses that condition unambiguously.

Head training. The backbone and frozen teacher are Qwen3-4B-Instruct-2507. Each batch sends the same 64 original requests to all four heads, producing 256 sampled responses with one response per request and head. The student receives only the request, without a persona card or survey preference. A Qwen3.5-27B critic receives the target persona card, request, and sampled response as one user message with thinking disabled. It returns a JSON object containing `judgement`, `changes needed`, and `feedback`. Only the feedback text conditions the teacher, while the judgment is logged and the change indicator controls the gate. The critic assesses persona fit and request fulfillment, not factual correctness.

The teacher receives the persona card as its system message and the original request followed by the feedback as its user message. It does not generate a replacement response. Instead, it scores the student's sampled continuation at every response position and supplies the top-256 targets used by the distillation objective. The complete student, critic, and teacher contexts appear in Appendix C.2. Responses contribute zero loss when the critic call fails, no feedback string can be parsed, or `changes needed` is `No`. Gated responses remain in the token denominator, as specified in Appendix A. Appendix F relates this gate to the reported training dynamics.

Table 5 gives the training configuration. Checkpoints are saved every 20 steps. We compare steps 40, 60, 80, and 100 on 50 randomly selected validation requests, evaluating each head against its own persona card, and use the step-60 heads for all reported evaluations.

Table 5: **Single-turn persona-head training configuration.** The backbone remains frozen, and the four heads are independent LoRA adapters.

Setting	Value
LoRA	Rank 32, scale 64, dropout 0, all linear layers
Training pass	6,400 original requests, 100 batches of 64
Batch sizes	Training 64, training microbatch 4, rollout microbatch 8
Student sampling	Temperature 0.7, top-p 1.0, one sample per head and request
Token limits	1,024 prompt tokens, 2,560 new response tokens
Teacher targets	Top-256 support, full-vocabulary student normalization
Optimizer	AdamW, learning rate 10^{-5}, $\beta = (0.9, 0.95)$
Schedule	Constant after 3% warmup, head learning-rate multiplier 1.0
Regularization	Weight decay 0, gradient-norm clipping 1.0
Critic decoding	Temperature 0, thinking disabled, at most 1,000 tokens
Training execution	bfloat16, DeepSpeed ZeRO-2, gradient checkpointing, SDPA attention
Rollout execution	vLLM, four engines with tensor parallelism 2, colocated with training
Hardware	Eight NVIDIA A100-SXM4-80GB GPUs, shared with the critic server
Random seed	7

Router. Router features are the frozen backbone's final-layer hidden state after the final normalization, at the last token of the chat-formatted rewritten request. This token is the newline following the assistant role marker in the generation prefix. Adapters are disabled when extracting the 2,560-dimensional features, which are standardized using training statistics. The router has layers $2{,}560 \to 512 \to 512 \to 4$, with ReLU activations and dropout 0.1 after each hidden layer. Its four sigmoid outputs are trained with binary cross-entropy against the assigned persona indicators in the order warm, impersonal, concise, and thorough. For example, warm plus concise has target $(1, 0, 1, 0)$.

The usable training rewrites are split by user into 5,123 fitting requests from 930 users and 554 development requests from 103 users. The optimizer is AdamW with learning rate 10^{-3}, weight decay 10^{-2}, and batch size 128. Early stopping monitors development binary cross-entropy with patience 15, selecting epoch 4. For each axis, the higher-scoring persona is selected only when its probability exceeds 0.5. If neither axis yields a selection, generation uses the base model. On test, the selected persona set matches the assigned set on 426 of 499 requests, including 222 of 247 paired requests.

One selected persona activates its head alone. Two selected personas use the signed composition coefficients in Appendix E.4, without generating separate candidate responses. For thorough pairs, the voice strength is selected from $\{1, 0.5, 0.25\}$ using the 50 thorough-pair validation requests. The checkpoint, router configuration, and composition coefficients are fixed before test evaluation. At deployment, the router and learned heads receive only the rewritten request, without a separately supplied condition label, survey preference, or persona card.

Evaluation generation. Responses are generated with vLLM using greedy decoding, at most 2,560 new tokens, maximum model length 4,096, and seed 42. The base and fixed-head baselines receive the same rewritten requests as PLF. The routed prompting baseline instead adds the selected persona card, or the two selected cards joined by one space, to the base model's system message. The pairwise judge receives the assigned condition label and both responses using the template in Appendix C.6. It is called with `max_completion_tokens`=2,000 and evaluates both presentation orders. Tie handling, preference win rates, and bootstrap intervals follow Appendix B.3. Single-turn bootstrap resampling uses seed 0.

B.2 Multi-Turn Tutoring Data, Training, and Routing

Data and simulated students. We use the Education Dialogue Dataset, which contains 8,000 Gemini-generated teacher–student dialogues across 77 topics (Shani et al., 2024). Filtering for dialogues that begin with a teacher turn, contain at least four turns, and carry one of four student-reaction annotations yields 5,350 candidates. After removing one topic with a corrupted name, we split by topic into 4,524 training dialogues across 65 topics, 200 validation dialogues across three topics, and 500 test dialogues across eight topics. The test topics span literature, biology, physics, politics, and history. Only the opening teacher turn is retained, with string-escaping artifacts removed. All subsequent student and tutor turns are generated, and the dataset's original preference and reaction annotations are not used.

Each simulated student is assigned a hidden preference for direct instruction or interactive guidance and a reaction style chosen from frustrated, rude, anxious, and disengaged. The reaction style remains fixed within a dialogue and shapes responses to unsatisfactory tutoring. The preference also remains fixed except in the switching evaluation. During head training, the reply to the dataset opening is preference-neutral. During router training and evaluation, the student expresses its assigned preference from that first reply onward. The student simulator uses Qwen3.5-27B with reasoning disabled. Its prompt templates are provided in Appendix D.3.

Head training. The backbone is Llama-3.1-8B-Instruct, with 83.9M trainable parameters per adapter. Each head trains against students whose hidden preference matches its intended behavior. Training consists of 100 rollout cycles. In each cycle, each head tutors 32 training openings for eight generated turns, yielding 256 tutor turns per head. Tutor turns are capped at 256 tokens. Batches contain 64 tutor turns, giving eight optimizer batches across the two heads per cycle. The learning rate is constant after the 3% warmup, and evaluation uses the final heads. A tutoring cycle therefore differs from a single-turn training batch in both the number of examples and the number of optimization steps.

The student's next reply supplies the feedback for the preceding tutor turn. The teacher treats this reply as a reaction to an earlier attempt and answers the original student message again, without receiving the sampled tutor response as a draft to copy. The exact framing is given in Appendix D.2.

Router. Router features are the final-layer hidden state at the last token of the chat-templated tutor context, computed with adapters disabled and standardized using training statistics. A three-layer MLP of width 512 predicts the student preference using the categorical cross-entropy objective in Eq. 3. The router always selects one of the two heads. Its data collection crosses both generating heads with both student types on 100 training openings, yielding 400 dialogues. Each dialogue contributes nine prefixes ending at student replies, for 3,600 labeled examples. Crossing heads and preferences prevents the generating head from determining the target label. The number of training epochs is selected using 20 held-out training openings, without using validation or test topics for fitting or epoch selection. The router is frozen before both fixed-preference and preference-switching evaluation.

Human calibration of the student simulator. Before fixing the simulator prompt, we use 50 held-out tutor–student contexts to check whether simulated reactions are consistent with the assigned direct-instruction or interactive-guidance preference. Three human annotators independently judge preference consistency, and we select the prompt configuration with the highest majority agreement. These examples are disjoint from the reported test set, and the simulator prompt is frozen before model training and evaluation.

B.3 Evaluation and Statistical Reporting

Policies and generation. We compare PLF with the frozen base, each persona head used as a fixed policy, and router-based persona prompting. The prompting baseline uses the same router as PLF but supplies the selected persona cards to the frozen base instead of activating learned adapters. The tutoring oracle activates the head matching the hidden preference. It is an information-privileged reference, not a deployable policy or a formal upper bound on dialogue quality.

Single-turn responses use greedy decoding. For tutoring, all policies receive the same test opening and the same initial student reply, while subsequent turns are sampled independently for each policy.

Each dialogue contains eight generated tutor turns at temperature 0.7, with at most 256 tokens per tutor turn and 128 tokens per student reply. Both student types use the same 500 test openings.

Pairwise preference fit. The preference judge, `gpt-6-luna`, supplies no training feedback. It receives the assigned preference and compares either two single-turn responses or two complete tutoring dialogues in both presentation orders. A win or loss requires the two orders to agree on the winner. Order disagreements, explicit ties, and identical single-turn responses count as ties. Switching evaluations also treat conflicting before- and after-switch verdicts as ties. For counts of wins W, ties T, and losses L, the reported preference win rate is $(W + T/2)/(W + T + L)$.

All reported confidence intervals are 95% percentile bootstrap intervals. Single-turn preference intervals use 4,000 resamples of requests. Fixed-preference tutoring intervals use 4,000 resamples of the 500 openings, retaining both student types together and weighting the types equally. Preference-switching win-rate intervals use 2,000 resamples of openings, retaining both switch directions together. Thus, the two types or directions for one opening are not treated as independent units in the aggregate preference intervals.

Tutoring satisfaction and factual errors. Student satisfaction is the fraction of replies to generated tutor turns that are not flagged as objections. The shared reply to the dataset opening is excluded. Factual-error rate is the fraction of generated tutor turns flagged by the fact-checking judge. These auxiliary judges use Qwen3.5-27B with reasoning disabled and greedy decoding. Their metrics are distinct from the pairwise preference judgment and need not rank policies identically. Satisfaction is a secondary measure because related judgments also contribute to training. The evaluation measures simulated preference satisfaction, not human learning outcomes.

For paired differences in tutoring satisfaction and factual-error rates, we use 2,000 bootstrap resamples of openings under fixed preferences and of opening–direction pairs under switching preferences. The switching satisfaction intervals likewise use 2,000 resamples of opening–direction pairs. Policies remain paired within each resampled unit. Single-turn preference results do not establish preservation of independently assessed factual correctness or task quality.

Human calibration of the preference judge. We calibrate the pairwise preference judge on 50 held-out response pairs labeled independently by three human annotators. We select the judge prompt with the highest agreement with the human majority judgment and freeze it before the reported test evaluation.

C Single-Turn Prompts

The templates below specify the persona cards, student, critic, distillation teacher, persona-prompting baseline, request rewriter, rewrite checks, and pairwise preference judge. Braced identifiers are fields filled at runtime, and prompt wording is retained. Student, teacher, and baseline contexts use the Qwen3-4B-Instruct-2507 chat template with an assistant generation prefix. The critic, rewriter, checks, and preference judge receive the corresponding user-message templates. Line wrapping is for readability and does not add instructions to the prompts.

C.1 Persona Cards

The four cards describe the two voice personas and two depth personas. Only the relevant card text is substituted for {`card`}, without the display headings below. The learned heads and router do not receive these cards at inference. The cards condition the training critic and teacher and are also used by the persona-prompting baseline.

Single-turn persona cards

Warm:
You're warm and relaxed: conversational, caring about how the person feels, and light-hearted when the moment suits.

Impersonal:
You're objective and professional: you answer without giving opinions of your own or taking sides, and you skip the greetings, compliments, small talk, feelings and moralising.

Concise:
You keep it short and plain: you get straight to the answer, in simple words, and stop there.

Thorough:
You go into depth: you explain the reasoning behind your answers, and when the user asks for a judgement or a choice, you set out more than one option or view and leave the decision to the person.

C.2 Student, Critic, and Distillation Teacher

Student. Each head samples a response to the original request without a system message containing its persona. The assistant prefix below is followed by the sampled response.

Student context

<|im_start|>user
{request}<|im_end|>
<|im_start|>assistant

Critic. The critic receives the persona card, request, and sampled response in a single user message. The request and response are stripped of leading and trailing whitespace before insertion. The `feedback` field conditions the teacher, while `changes needed` controls the no-change gate. The `judgement` field is logged but is not passed to the teacher.

Single-turn critic: user message

Assistant persona
{card}

User request
{request}

Assistant response
{response}

Judge how well the response fits the persona while still giving the user what they asked for: whatever they ask for (a joke, a story, a plan, a recipe, a choice, a hypothetical) is given in full, whatever the persona. Then give the assistant feedback it can learn from for requests like this one: what to leave out, anything that doesn't fit the persona, and which of its information and advice to keep, in your own words, without writing sentences for it to copy or mentioning these instructions. Don't comment on whether its facts are right, and don't ask for extra examples, details or more of the persona.

Reply with a JSON object:
{"judgement": "Your judgement", "changes needed": "Yes or No", "feedback": "Your feedback to the assistant"}

Distillation teacher. The frozen teacher receives the same request with the target card and response-specific feedback. Only the critic's feedback string is inserted, not its full JSON response. The teacher scores the student's continuation token by token under this context rather than generating another answer. The student's response is therefore present as the continuation being scored, not as an additional draft in the teacher's instructions.

Single-turn teacher context

<|im_start|>system
{card}<|im_end|>
<|im_start|>user
{request}

The following is future feedback you received. Use this to guide your answer to the user prompt:

{feedback}

Apply it without mentioning it, apologising, or saying what changed.<|im_end|>
<|im_start|>assistant

C.3 Persona-Prompting Baseline

For a single selection, {card} is the selected persona card. For a paired selection, it is the voice card followed by one space and the depth card. For the routed baseline, these selections come from the same router used by PLF, rather than from the evaluation label. The request field contains the preference-embedded evaluation request.

This baseline serves as the direct response-independent reference for persona conditioning. It exposes the frozen backbone to the same persona specification that conditions the PLF teacher, but without the response-specific feedback used during PLF training. Therefore, it tests whether the learned adapters improve over directly eliciting the intended persona from the same frozen model.

Persona-prompting baseline context

<|im_start|>system
{card}
<|im_end|>
<|im_start|>user
{request}<|im_end|>
<|im_start|>assistant

C.4 Preference-Embedded Request Construction

The rewriter receives the original request as {prompt} and one assigned condition label as {preference}. The labels are `likes warm, friendly answers`, `doesn't want the assistant's own opinions`, `likes short answers`, and `likes in-depth answers`. A paired label joins the voice label and depth label with `, and `, including the separating spaces. For example, the warm–concise label is `likes warm, friendly answers, and likes short answers`. The same condition label is later supplied to the preference judge, while the response policy and router see only the rewritten request.

Preference-embedded request rewriter: user message

Here is a message a user typed to a chat assistant:

<message>
{prompt}
</message>

This user {preference}. Minimally rewrite the message so that this preference comes through naturally, the way this user would really have typed it.

- Keep the request exactly the same, and keep their own words, spelling and casing wherever you can.
- Make the smallest change that does it: a different way of wording it, or a few casual words of their own. Don't restate or explain the preference, and add no backstory.
- If the user has two preferences, both must come through.
- This is for a dataset of realistic user messages: don't answer or judge the message.

Reply with only the rewritten message.

C.5 REWRITE CHECKS

The reader check receives only the rewritten request and predicts the two preference axes independently, allowing `none` on either axis. The same-task check receives both versions and assesses whether the rewrite changes the requested task rather than only its presentation. These checks are recorded for inspection and are not used to filter the training, validation, or test sets. They supply neither distillation feedback nor additional router labels.

Preference reader check: user message

Below is a message that a user sent to a chat assistant. Some messages give away how the user likes to be answered, through who they are, their situation or the way they write; many give nothing away.

<message>
{prompt}
</message>

From the message alone, judge two separate axes.

voice:
- "warm": they would like a friendly, relaxed, caring answer, with a bit of humour when it fits.
- "impersonal": they want no opinions from the assistant and no taking sides; professional, with no small talk and no pretending to be human.
- "none": nothing in the message points either way.

depth:
- "concise": they want a short, plain answer that gets straight to the point.
- "thorough": they want depth and the reasoning behind the answer, and more than one option or view when there is a choice to make.
- "none": nothing in the message points either way.

Answer "none" unless something in the message gives you a reason: what the user says about themselves or their situation, or the way they write. The topic alone is not a reason, and neither is an ordinary short question.
Reply with one line of JSON and nothing else: {"voice": "warm" | "impersonal" | "none", "depth": "concise" | "thorough" | "none"}

Same-task check: user message

Here are two versions of a message a user sent to a chat assistant.

<original>
{original}
</original>

<rewritten>
{rewrite}
</rewritten>

Does the rewritten version still ask for the same thing? These are NOT changes: different wording, tone or length; a greeting or thanks; and a wish about how the answer should be (short, in depth, friendly, without the assistant's opinions). It is CHANGED only if what is being asked differs: a different question or claim, a request added or dropped, or facts, constraints or stance changed (for example "I'm not sure I support X" becoming "I don't support X").

Reply with one line: SAME or CHANGED, then a colon and a short reason.

C.6 PAIRWISE PREFERENCE JUDGE

The evaluation judge receives the assigned condition label, rewritten request, and two generated responses. Each call requests an A-or-B verdict, and the pair is evaluated again with the response order reversed. An overall win or loss requires agreement between the two orders, while disagreement or identical responses produces a tie. This prompt evaluates preference fit and does not provide an independent factual-accuracy score.

Single-turn pairwise preference judge: user message

Two assistants each answered the same request from a user with this preference for how an assistant should reply:
{preference}

Request: {prompt}

=== Response A ===
{response_a}

=== Response B ===
{response_b}

Which assistant fits this user's preference better?

Reply in exactly this format:
REASON: <one or two sentences>
VERDICT: A | B

D MULTI-TURN TUTORING PROMPTS

The following templates specify the tutor, distillation teacher, student simulator, and judges used in the tutoring experiments. The prompt wording is retained, with braced identifiers denoting fields filled at runtime. In the tutor's context, the dataset opening and generated tutor turns are assistant messages, and student replies are user messages. The simulator uses the opposite role assignment. The router receives the tutor's context without persona descriptions.

D.1 TUTOR AND PERSONA DESCRIPTIONS

Every tutor policy, including the learned heads, receives the common tutor system prompt below. The learned heads do not receive persona descriptions. The descriptions are used by the distillation teacher and persona-prompting baselines, where they are inserted after the first sentence of the common system prompt.

Tutor system prompt

You are a teacher in a tutoring conversation about {topic}.

Talk like a real tutor: one short reply at a time, teaching a single small thing per turn — keep it brief and conversational, not a long monologue, list, or lesson plan. Teach accurately and react to what the student just said. Only write the next Teacher turn.

Persona descriptions

Direct instruction: Your teaching style is direct instruction: you clearly explain and lay the material out for the student.

Interactive guidance: Your teaching style is interactive back-and-forth discussion: you ask the student questions and reason out the answers together.

D.2 DISTILLATION TEACHER

The teacher is the frozen base model with the target persona description inserted into its tutor system prompt. The dialogue preceding the sampled tutor response remains unchanged. The student's subsequent reply is appended to the system prompt as a reaction to an earlier attempt, while the sampled tutor response itself is omitted. This framing instructs the teacher to answer the original student message rather than continue the conversation from the reaction.

Teacher context (appended to the system prompt)

The student's reaction to an earlier attempt at your reply to their latest message:

{student_reply}

Reply to the student's latest message again, using this reaction. Do not mention the reaction, quote it, apologise for the earlier attempt, or refer in any way to having been given it. Write the reply as though it were your first.

D.3 Student Simulator

The field {student_reaction} is filled with frustrated, rude, anxious, or disengaged and remains fixed within each dialogue. During head training, the simulator uses the neutral prompt for its reply to the dataset opening and the preference-conditioned prompt for subsequent replies. During router training and evaluation, it uses the preference-conditioned prompt from the first reply. In the preference-switching experiment, the prompt changes to the new preference beginning with the reply to the switch turn.

Student: reply to the opening during head training (neutral)

You are a student in a tutoring conversation about {topic}. The teacher has just opened the lesson. Write your first reply the way a real student would — just react to the opening, or answer it if it asked you something, without asking for anything or saying how you like to learn. Usually one short sentence.

Student: direct-instruction preference

You are a student in a tutoring conversation about {topic}. You like being told things clearly. What counts is what the teacher's last turn actually did: if it explained or told you something new, you're into it and take it in; if it put you on the spot to answer, you get {student_reaction}; if it didn't teach you anything new — just praise, filler, or saying again what you already had — you get impatient and complain that it didn't tell you anything new. If it said something you know is wrong, you point that out. React to that turn only, not to the lesson so far. Write only your next reply, the way a real student talks — usually one short sentence.

Student: interactive preference

You are a student in a tutoring conversation about {topic}. You like being asked questions and working out the answers yourself, not being told them or asking for them. What counts is what the teacher's last turn actually did: if it asked you something about {topic} worth thinking about, you're into it and take a stab at the answer; if it answered something you asked, that's fine; if it just told you things you didn't ask about, you get {student_reaction} and say you'd rather have been asked. If its question didn't really make you think — just asking how you feel or what stands out to you, or something you'd already answered — you get impatient and complain that it wasn't a real question. If it said something you know is wrong, you point that out. React to that turn only, not to the lesson so far. Write only your next reply, the way a real student talks — usually one short sentence.

D.4 Training and Evaluation Judges

Training gates. The objection judge identifies complaints, requests for a different teaching approach, and corrections in the student's reply. It sees only the tutor turn and student reply, without the hidden preference or reaction style. The satisfaction judge separately checks for evidence that the student engaged with the turn. A turn is masked only when the satisfaction judge accepts it, the objection judge does not flag it, and the question-back check is also satisfied. The complete gating rule is described in Appendix F.

Evaluation metrics. The evaluation objection prompt defines the satisfaction metric. The fact-checking judge receives the topic and one tutor turn, while the preference judge receives the topic, the preference description, and both complete dialogues rendered as `Teacher:`/`Student:` lines.

The pairwise template below is used for fixed-preference dialogues. For switching evaluation, the judge is instead given both preference descriptions and the marked switch point in each dialogue. Each pair is judged in both presentation orders, as detailed in Appendix B.3.

Objection judge: training (user message)

Teacher's turn:
{tutor_turn}

Student's reply:
{student_reply}

Did the student want the teacher's turn to be different? Answer yes if they complain, push back, ask to be taught another way ("just tell me", "ask me instead"), or say or suggest the teacher got something wrong, even while answering or asking a question. Answer no if they answer, ask a follow-up question about the topic, or agree. Answer only yes or no.

Objection judge: evaluation (user message)

Teacher's turn:
{tutor_turn}

Student's reply:
{student_reply}

Did the student want the teacher's turn to have been different? Answer yes if the student complains, pushes back, or asks the teacher to handle it another way (for example "just tell me" after being asked something, or "ask me" after being told something). Answer yes also if the student says something in the teacher's turn is wrong, even if it also answers. Answer no if the student answers, asks a follow-up question about the topic, or agrees. Answer only yes or no.

Satisfaction judge (user message)

Teacher's turn:
{tutor_turn}

Student's reply:
{student_reply}

Was the student really satisfied with the teacher's turn? Answer yes only if the reply shows it: the student takes in something new the teacher told them, saying it in their own words; takes a real stab at the teacher's question (a guess is fine, even phrased as a question); or builds on the turn with an idea of their own. Answer no if the student complains, pushes back, corrects or doubts anything the teacher said or asked, or is confused; asks the teacher something back instead ("So X?", or asking it to explain, finish or say more); only acknowledges ("Okay.", "That makes sense."); or says they don't know. Answer only yes or no.

Fact-checking judge (user message)

A teacher in a tutoring conversation about {topic} said:

{tutor_turn}

Does this contain a factual error? Ignore opinions, questions and simplifications a teacher would reasonably make; only flag something that is actually false. Answer only yes or no.

Pairwise fit judge (user message)

Two tutors each taught the same topic to a student with this learning preference:
{preference}

Topic: {topic}

=== Conversation A ===
{dialogue_a}

=== Conversation B ===
{dialogue_b}

Which tutor fits this student's learning preference better? Judge from the student's reactions: read each student reply against the tutor turn before it. Complaints, pushback and requests to be taught differently count against the tutor, even if it adapts afterwards. Engaged, satisfied replies count for it.

Reply in exactly this format:
REASON: <one or two sentences>
VERDICT: A | B

Preference descriptions for the fit judge (filled into {preference})

Direct instruction: direct instruction: being told and shown things clearly, not quizzed or asked to guess

Interactive guidance: interactive back-and-forth: being asked questions and reasoning the answers out, not lectured or asked empty questions like "what stands out to you?"

E Supplementary Preference Results

This section provides detailed results for the preference-learning, composition, and preference-switching evaluations in Section 4. Win rates give half credit to ties, and intervals follow the re-sampling procedures in Appendix B.3.

E.1 Overall Preference Fit

Table 6 provides the counts and intervals underlying Figure 2. Single-turn results use 499 requests. Tutoring aggregates 500 openings evaluated with each of the two student types. The 1,000 tutoring comparisons share 500 openings, which remain paired when computing confidence intervals.

Table 6: **Preference fit with full judgment counts.** Wins, ties, and losses are from PLF's perspective. Win rates and confidence intervals are percentages. The tutoring oracle knows the hidden preference.

PLF versus	n	Wins	Ties	Losses	Win rate [95% CI]
Single-turn personalization					
Warm head	499	341	95	63	77.9 [75,81]
Impersonal head	499	337	100	62	77.6 [74,81]
Concise head	499	323	113	63	76.1 [73,79]
Thorough head	499	328	115	56	77.3 [74,80]
Base	499	280	79	140	64.0 [60,68]
Router + persona prompt	499	229	89	181	54.8 [51,59]
Multi-turn tutoring, both student types					
Direct head	1,000	555	143	302	62.7 [60.2,65.2]
Interactive head	1,000	634	118	248	69.3 [67.0,71.6]
Base	1,000	589	107	304	64.3 [61.6,67.0]
Router + persona prompt	1,000	494	118	388	55.3 [52.6,58.2]
Oracle	1,000	384	182	434	47.5 [44.8,50.3]

Setting	n	PLF wins	Ties	Prompt wins	PLF win rate
Single-turn	96	43	22	31	56.3%
Multi-turn tutoring	100	47	18	35	56.0%

Table 7: Human preference evaluation of PLF against routed persona prompting.

E.2 Human Evaluation

To validate our model-based preference evaluation, we conduct a blind human evaluation comparing PLF with routed persona prompting, which uses the same router as PLF. Three volunteers independently evaluate all sampled comparisons. All participants provide informed consent. The study was approved by the relevant Institutional Review Board (IRB) before data collection.

We randomly sample 96 single-turn comparisons, balanced across the eight preference conditions, and 100 tutoring comparisons, balanced across the two learning preferences. Participants see the target preference, request or dialogue, and two anonymized outputs in randomized order, and select which better fits the stated preference while appropriately addressing the request, with *Tie* also allowed.

We aggregate the three judgments by majority vote and report win rates with ties receiving half credit. Human majority judgments agree with the model judge on approximately 76% of single-turn comparisons and 73% of tutoring comparisons, and show the same overall preference for PLF over routed persona prompting.

E.3 Multi-Turn Preference Fit and Tutoring Quality

Table 8 separates the tutoring comparisons by student type. PLF's preference advantage over routed persona prompting is similar for direct-instruction and interactive students, at 55.5% and 55.1%, respectively. A fixed head that matches the hidden preference is a stronger comparator. For direct-instruction students, PLF has a 44.4% win rate against the direct head, compared with 80.9% for interactive students. This distinction explains why an aggregate advantage over fixed policies can coexist with routing errors for a particular student type.

Table 8: **Tutoring preference fit by student type.** Each comparison uses the same 500 test openings. Counts are PLF wins, ties, and losses against the indicated policy. Win rates and confidence intervals are percentages.

PLF versus	Wins	Ties	Losses	Win rate [95% CI]
Direct-instruction students				
Direct head	173	98	229	44.4 [40.6,48.4]
Interactive head	432	32	36	89.6 [87.2,91.9]
Base	246	56	198	54.8 [50.7,58.8]
Router + persona prompt	244	67	189	55.5 [51.5,59.6]
Oracle	173	97	230	44.3 [40.4,48.3]
Interactive students				
Direct head	382	45	73	80.9 [77.7,84.0]
Interactive head	202	86	212	49.0 [45.0,53.1]
Base	343	51	106	73.7 [70.0,77.3]
Router + persona prompt	250	51	199	55.1 [50.9,59.2]
Oracle	211	85	204	50.7 [46.8,54.6]

Table 9 reports satisfaction and factual-error rates for the same policies. Compared with routed prompting, PLF increases satisfaction from 79.8% to 82.1%, a paired gain of 2.3 percentage points with 95% CI [1.1,3.4]. The gain is 3.5 points [1.9,5.2] for direct-instruction students and 1.1 points [-0.5,2.7] for interactive students. PLF's aggregate satisfaction is 6.4 points [5.2,7.6] above the base and 0.6 points below the oracle, with a paired difference of -0.6 [-1.5,0.3].

The factual-error judge flags 7.1% of PLF turns, compared with 8.1% for routed prompting and 7.9% for the base. The paired difference is -1.0 percentage points [-2.0,-0.1] against routed prompting

and -0.8 [-1.6,0.1] against the base. Thus, fewer errors are flagged relative to routed prompting, while the difference from the base is less certain. These are model-judged tutoring outcomes rather than a general assessment of factual reliability.

Table 9: **Student satisfaction and factual errors.** Each policy generates eight tutor turns on 500 openings for each student type. D and I denote direct-instruction and interactive students. Satisfaction is the percentage of replies not flagged as objections, and factual errors are the percentage of tutor turns flagged by the fact-checking judge. Means weight the student types equally. The oracle uses the matching head, with differences from the corresponding fixed-head run due to sampling.

	Satisfaction ↑			Factual errors ↓		
Policy	D	I	Mean	D	I	Mean
Base	77.8	73.5	75.6	9.2	6.5	7.9
Direct head	81.9	68.3	75.1	9.0	5.6	7.3
Interactive head	62.3	84.1	73.2	7.7	5.6	6.7
Router + persona prompt	77.1	82.5	79.8	10.0	6.2	8.1
PLF	80.6	83.5	82.1	9.3	4.9	7.1
Oracle	81.9	83.4	82.7	9.1	5.8	7.4

E.4 Single-Turn Composition

Validation-selected coefficients. For a requested voice persona V and depth persona D, with opposites $\bar{V}$ and $\bar{D}$, the signed composition rule in Eq. 6 uses

$$(w_V, w_{\bar{V}}, w_D, w_{\bar{D}}) = \left(\frac{1}{4} + \frac{\lambda_V}{2}, \frac{1}{4} - \frac{\lambda_V}{2}, \frac{1}{4} + \frac{\lambda_D}{2}, \frac{1}{4} - \frac{\lambda_D}{2}\right). \tag{12}$$

Concise pairs use $\lambda_V = \lambda_D = 1$. Thorough pairs use $\lambda_V = 0.25$ and $\lambda_D = 1$, with the weaker voice strength selected on validation data. Thus, composition can subtract the opposite persona's update as well as add the requested one. The resulting weight sum is implemented exactly through factor concatenation, as detailed in Appendix A.

Table 10 reports the validation comparison discussed in Section 4.3. The 100 paired-preference validation requests contain 25 examples of each pair, and composition uses the assigned pair labels rather than predicted routes. All rules are compared against the frozen base prompted with both requested persona cards. The sum assigns coefficient one to each requested head, while the average assigns coefficient one half to each. Because these requests also select the voice strength, the comparison documents configuration selection rather than independent test evidence.

Table 10: **Validation comparison of composition rules.** Win rates against prompting with both persona cards, with half credit for ties. Rates and confidence intervals are percentages.

Composition rule	Win rate	95% CI
Sum of the requested heads	39	[30,48]
Average of the requested heads	50	[42,59]
Axis-weighted composition	70	[62,78]

Held-out composition results. Table 11 expands the correctly routed comparisons in Table 1. The requested pair is selected correctly on 222 of the 247 paired-preference test requests. Across these 222 requests, the mixture's pooled win rate is 79% [74,84] against the requested voice head and 67% [62,72] against the requested depth head. Warm–concise benefits most from combining both behaviors, while impersonal–concise remains close to the concise head.

Table 12 includes routing errors in the constituent-head comparisons discussed in the main text. Including all paired requests reduces the warm–concise win rates against its constituent heads from 93% and 92% to 81% and 76%. The impersonal–concise configuration remains near parity with the concise head at 49%. The pair-prompting comparisons in Section 4.3 also show that both thorough mixtures remain below prompting with both persona cards, despite outperforming their constituent

Table 11: **Composition on correctly routed test requests.** Mixture win rates (%) against each fixed head, with 95% confidence intervals. Bold rates compare with a constituent of the requested pair. Results cover 222 correctly routed requests out of 247 paired-preference requests.

Requested pair	n	Warm	Impersonal	Concise	Thorough
Warm + concise	48	**93** [85,98]	94 [86,99]	**92** [84,98]	99 [97,100]
Warm + thorough	59	**88** [81,94]	94 [88,98]	98 [96,100]	**68** [57,78]
Impersonal + concise	55	80 [70,89]	**60** [48,72]	**48** [38,59]	81 [71,90]
Impersonal + thorough	60	89 [82,95]	**75** [64,84]	85 [77,92]	**64** [55,73]

heads. These comparisons distinguish the benefit of composition from the quality of preference selection and of the individual learned heads.

Table 12: **Composition including routing errors.** PLF win rates against the requested voice and depth heads on all requests assigned to each pair, including routing errors. Correct n gives the number whose selected pair matches the assigned pair. Win rates are percentages with half credit for ties and use all requests in the row as the denominator.

			PLF versus	
Requested pair	All n	Correct n	Voice head	Depth head
Warm + concise	62	48	81	76
Warm + thorough	62	59	85	65
Impersonal + concise	61	55	61	49
Impersonal + thorough	62	60	73	64

E.5 Within-Session Preference Switching

Protocol and preference fit. For each test opening, the hidden preference switches once in each direction at a turn j sampled uniformly from $\{2, \ldots, 7\}$. Both directions for an opening use the same switch turn. The student first reacts under the new preference in its reply to turn j, without announcing the change. PLF can therefore first respond to evidence of the switch at turn $j + 1$. The oracle instead knows the preference that will evaluate each turn and switches at j. All other policies have the same information access as in the fixed-preference evaluation.

Table 13 expands Table 2. The mean averages the two directions on the same 500 openings and is not an additional independent group. PLF is preferred to both fixed heads and the base in aggregate, but remains below the oracle, especially for switches toward direct instruction.

Table 13: **Preference fit under switching.** PLF win rates (%) with 95% confidence intervals. Each direction uses 500 test openings. Bold rates compare with the fixed head matching the preference before the switch. The oracle knows the new preference when choosing turn j.

Switch	Direct head	Interactive head	Base	Oracle
Direct → interactive	**55** [51,58]	81 [78,83]	54 [51,58]	47 [43,51]
Interactive → direct	67 [64,71]	**56** [52,60]	59 [55,62]	40 [36,44]
Mean	61 [59,63]	68 [66,70]	56 [54,59]	43 [41,46]

Satisfaction around the switch. Tables 14 and 15 provide the numerical results corresponding to Figure 3, including the breakdown by direction. The aligned buckets summarize turns before j, at j, at $j + 1$, at $j + 2$, and at or after $j + 3$. Within a direction, each bucket pools the available turns, and the aggregate averages the two direction-specific percentages. Later buckets contain only dialogues with enough remaining turns, so the buckets have unequal support and are not a fixed-cohort longitudinal comparison. PLF's satisfaction initially recovers after the switch but does not consistently exceed the strongest fixed head in later buckets.

Table 14: **Satisfaction under switching, averaged over directions.** All satisfaction values are percentages. The overall column includes a 95% confidence interval over opening–direction pairs. The remaining columns align tutor turns to the switch at j and average the direction-specific bucket percentages.

		Satisfaction around the switch				
Policy	Overall [95% CI]	Before j	j	$j+1$	$j+2$	$\geq j+3$
Base	73.0 [72.0,74.1]	77.4	64.1	73.7	69.0	70.3
Direct head	71.4 [70.4,72.5]	74.9	62.7	73.7	65.1	71.0
Interactive head	66.4 [65.3,67.6]	69.5	57.8	67.9	64.0	65.2
PLF	76.7 [75.7,77.6]	84.9	62.5	74.3	71.8	70.7
Oracle	79.2 [78.3,80.2]	86.4	70.8	74.7	73.4	74.0

Table 15: **Switching satisfaction by direction.** Satisfaction percentages over 500 dialogues per direction, with buckets aligned to the switch turn j. The oracle anticipates the switch, while PLF first observes evidence of it after generating turn j.

Policy	Overall	Before j	j	$j+1$	$j+2$	$\geq j+3$
Direct instruction → interactive guidance						
Base	77.1	81.0	69.4	77.8	75.9	73.3
Direct head	79.1	85.9	72.6	76.4	72.5	72.5
Interactive head	61.3	52.6	62.0	69.6	70.4	70.9
PLF	79.8	84.9	69.6	79.2	79.0	75.1
Oracle	81.3	87.0	73.2	78.6	77.6	77.3
Interactive guidance → direct instruction						
Base	69.0	73.9	58.8	69.6	62.2	67.4
Direct head	63.8	63.8	52.8	71.0	57.6	69.4
Interactive head	71.5	86.3	53.6	66.2	57.6	59.6
PLF	73.6	85.0	55.4	69.4	64.6	66.4
Oracle	77.1	85.8	68.4	70.8	69.2	70.7

Detection delay. Table 16 reports when PLF selects the newly matching head. It does so at the first opportunity in 39.8% of dialogues, with a higher rate for switches toward interactive guidance than toward direct instruction. In 338 of 1,000 dialogues, it does not select the new head before the lesson ends. Among dialogues in which it does, the mean delay beyond the first opportunity is 0.65 turns. This conditional delay excludes dialogues without a detected switch, so it should be read together with the no-detection count. The router sees the complete history despite having trained only on fixed preferences. Continued support for the old preference in earlier turns is a plausible explanation for the lag, but these diagnostics do not establish its cause.

Table 16: **PLF's detection of a preference switch.** First opportunity is the percentage of dialogues selecting the new head at $j+1$. Additional delay counts turns beyond $j+1$ until the first selection of the new head, conditional on that selection occurring. No detection counts dialogues ending without such a selection.

Switch	First opportunity (%)	Additional delay	No detection
Direct → interactive	48.0	0.63	113/500
Interactive → direct	31.6	0.67	225/500
Both directions	39.8	0.65	338/1,000

E.6 Qualitative Example Provenance

Table 3 uses single-turn request `c3424` and tutoring dialogues 1189 and 743 from the held-out evaluations. For the guitar request, the mixture is preferred to both constituent heads in both presentation orders. Dialogue 1189 switches from direct instruction to interactive guidance at turn 6, and dialogue 743 switches from interactive guidance to direct instruction at turn 2. Both tutoring examples were selected from successful first-opportunity switches: the first reaction under the new preference is flagged as an objection, while the reaction to the next tutor turn is not. They illustrate immediate adaptation when it succeeds, rather than estimating how often it occurs. The quoted wording is preserved with normalized typography, added emphasis, and omission of the guitar emoji.

F Feedback Gating and Training Dynamics

Single-turn gating. A response contributes no distillation loss when the critic reports that no changes are needed or provides no usable feedback. Otherwise, the critic's feedback conditions the teacher target on the sampled response prefixes. The no-change rate in Figure 4a tracks critic approval during training. It is not a held-out measure of preference fit or factual quality.

Tutoring gating. A tutor turn is masked only when the satisfaction judge accepts the student's reply, the objection judge does not flag it, and the student does not ask a question back after a question-free tutor turn. The last check tests whether the student reply ends in `?` while the tutor turn contains no question mark. This prevents an otherwise accepted checking question from automatically removing the training signal. A failed judge call leaves the turn eligible for training, while a simulator failure leaves the turn out of the distillation loss.

Three quantities therefore have different meanings: training satisfaction is acceptance by the satisfaction judge, the masking rate additionally requires the objection and question-back checks to pass, and evaluation satisfaction is the absence of a flagged objection. Figure 4b plots the first quantity, not the masking rate or the evaluation metric.

Loss normalization. Valid response tokens from masked examples remain in the denominator of Eq. 2. Masking therefore changes both which responses contribute supervision and the amount of active loss relative to the batch's total token count. Batches with no positive update gates are skipped. The observed training curves describe the gated procedure, rather than isolating the effect of the gate from the rest of training.

F.1 Tutoring Training Summaries

Table 17 summarizes the training satisfaction curve and masking rates in blocks of 20 cycles. For direct instruction, satisfaction rises from 68.0% in the first block to 70.3% in the second and ends at 69.9%. Interactive guidance rises from 67.9% to 73.5% before ending at 71.8%. Masking remains near 49% for direct instruction and reaches about 67% for interactive guidance. These summaries complement the cycle-level curves without treating training judgments as evidence of held-out quality.

Table 17: **Tutoring feedback and masking during training.** Percentages are averaged within blocks of 20 rollout cycles. Satisfied denotes acceptance by the training satisfaction judge. Masked denotes responses excluded from distillation after all gating checks. Each cycle contains 256 tutor turns per head.

	Direct-instruction head		Interactive head	
Cycles	Satisfied	Masked	Satisfied	Masked
1–20	68.0	48.4	67.9	60.1
21–40	70.3	48.9	73.5	67.8
41–60	71.0	50.7	72.9	66.9
61–80	70.6	48.8	72.0	67.6
81–100	69.9	49.0	71.8	66.9

G No-Change Mask Ablation

G.1 Single-turn: fixed-router comparison

We compare the gated run (`run21`) with an ungated run (`run23`) at step 60 on the same 499 held-out requests. Both use the same backbone, persona cards, critic and teacher prompts, LoRA configuration, optimizer schedule, and nominal rollout budget. Disabling the no-change mask trains on approved responses using their written feedback; unparseable critic replies are still excluded. The router is frozen and shared, and both systems decode greedily with the same response cap. Pairwise judging follows Appendix B.3. Table 18 reports the gated system's win rates; preference

rows use the assigned request condition, while the one- and two-pole rows group by the router's actual selection.

Table 18: **Single-turn gate ablation at step 60.** Gated versus ungated PLF on test requests. Win rates (%) give ties half credit; intervals are 95% percentile bootstraps over requests. Two base-fallback requests are excluded from the router-selection groups.

Group	n	W/T/L	Win rate	95% CI
All requests	499	205/151/143	**56**	[53,60]
Router selects one pole	232	92/70/70	55	[49,60]
Router selects two poles	265	112/80/73	57	[52,62]
Warm	63	29/19/15	61	[51,71]
Impersonal	63	29/12/22	56	[44,67]
Concise	63	19/24/20	49	[40,59]
Thorough	63	15/20/28	40	[30,49]
Warm + concise	62	28/15/19	57	[47,68]
Warm + thorough	62	31/16/15	63	[52,73]
Impersonal + concise	61	28/24/9	66	[56,75]
Impersonal + thorough	62	26/21/15	59	[48,69]

Critic approval differs little between runs over steps 1–60: gated versus ungated means are 59/57% for warm, 42/44% for impersonal, 33/33% for concise, and 28/27% for thorough. Yet the held-out preference outcomes differ. Judge rationales describe ungated warm answers that substitute emotional reassurance for the requested information, consistent with sampled approval feedback encouraging emotional connection. Thorough-only requests favor the fuller ungated answers, suggesting that the same additional supervision can benefit a less developed persona. These are qualitative interpretations of the recorded feedback and judgments, not an isolated test of the mechanism.

Several controls limit the interpretation. Step 60 was selected on validation for the gated run and matched for the ungated run, which stopped after step 61. Mixture weights were selected using gated heads and reused unchanged, potentially favoring that system on paired preferences. Even with the same seed, the training prompt orders differed, so equal step counts do not imply identical examples seen. Each variant has one training run. Masking also changes active loss scale: only 25–50% of tokens per head carry loss over steps 41–60 in the gated run, versus essentially all tokens without the mask. Separating response selection from loss scaling would require a further normalization control.

G.2 TUTORING: MATCHED-HEAD COMPARISON

We compare the final heads from `t9` and its no-mask variant after 100 cycles, using the same anchored teacher framing and training configuration. Each of 500 openings from eight held-out topics is evaluated with both student types. Students are assigned their matching head in advance, so this comparison excludes routing error. The paired eight-turn lessons share their first student reply and sampling seeds. The ungated run retains both training judges for logging, while every valid turn carries loss. The separate `gpt-6-luna` judge evaluates each dialogue pair in both orders.

Table 19: **Tutoring gate ablation with matched heads.** Gated versus ungated heads. Win rates (%) use half credit for ties. The two-sided sign test compares wins with losses, excluding ties; the pooled row combines both student types on the same 500 openings.

Student preference	n	W/T/L	Win rate	Sign-test p
Direct instruction	500	215/73/212	50.3	0.92
Interactive guidance	500	242/70/188	55.4	0.011
Both types	1,000	457/143/400	52.9	0.056

The preference advantage is concentrated in interactive guidance (Table 19). The gated head asks questions more often and draws fewer repetition complaints, while students more often satisfy the full engagement criterion. Supporting measures in Table 20 agree with this interpretation. Direct-

instruction comparisons provide little evidence of a difference. The pooled sign test is also less decisive than the interactive comparison.

Table 20: **Tutoring behavior under gated and ungated training.** Rates per 100 tutor turns; differences are gate minus no gate, with 95% paired bootstrap intervals over opening–student pairs. Strict satisfaction uses the full three-part gate rule; no objection is the evaluation satisfaction metric. Rates and differences are rounded independently.

Measure	Gate	No gate	Difference [95% CI]
Interactive: strict satisfaction	74.5	70.5	$+4.0$ [1.7,6.4]
Interactive: no objection	85.1	82.7	$+2.4$ [0.9,4.0]
Interactive: asks a question	86.3	82.4	$+3.9$ [1.7,6.1]
Interactive: repetition complaints	2.7	4.2	-1.5 [-2.4,-0.7]
Direct: strict satisfaction	40.4	39.0	$+1.4$ [-1.3,3.8]
Direct: no objection	81.9	81.1	$+0.8$ [-0.7,2.4]
Direct: student corrections	6.3	6.3	-0.1 [-1.1,0.9]

During training, the gated interactive head leads by 3.8 points [2.3,5.3] in strict satisfaction over cycles 16–50; the two runs converge on this training measure after cycle 50. The held-out preference and satisfaction advantages therefore persist beyond that early training gap. Validation differences are inconclusive on the three-topic split. Student correction rates change little, and the preference judge does not independently establish a factual-quality gain. Students and gate judges share the Qwen3.5-27B model family, and each variant has only one training run. Masking retains rollout and teacher computation: wall-clock times are nearly identical (5 h 19 min versus 5 h 20 min).

H KL-Direction Ablation

H.1 Protocol and preference fit

We compare the single-turn forward-KL run (`run21`) with its reverse-KL counterpart (`run22`) at step 60. Both retain the same backbone, persona cards, feedback source, no-change mask, and training recipe, with the sparse loss variants described in Appendix F. Each request is answered greedily by its assigned persona head, without routing or composition. The 500 test requests are split into four disjoint groups of 125 and minimally rewritten for the assigned persona. One refused impersonal rewrite leaves 499 pairs. The `gpt-6-luna` judge uses the persona label and both presentation orders; only agreement yields a decisive win or loss, and ties receive half credit.

Table 21: **Forward versus reverse KL at step 60.** Forward-KL win rates (%), with 95% bootstrap intervals over requests. The disjoint assignment uses each original request once; the expanded evaluation rewrites each test request for every persona (about 500 per persona). Its rates are converted from the reported reverse-KL perspective. The expanded evaluation reuses the original tasks; no pooled estimate is reported for it.

	Disjoint assignment			All prompts per persona
Persona	n	W/T/L	Win rate [95% CI]	Win rate [95% CI]
Warm	125	34/48/43	46 [40,53]	48 [45,52]
Impersonal	124	54/29/41	55 [48,63]	59 [56,63]
Concise	125	53/43/29	60 [53,66]	51 [48,55]
Thorough	125	37/33/55	43 [36,50]	40 [36,43]
All	499	178/153/168	51 [47,55]	—

The pooled result is near parity, masking variation across personas (Table 21). Concise favors forward KL in the disjoint assignment, but this advantage does not persist when all tasks are rewritten for that persona. The expanded evaluation instead gives the clearest evidence for forward KL on impersonal responses and reverse KL on thorough responses. Warm remains inconclusive. Figure 5 shows which effects persist across the two request assignments.

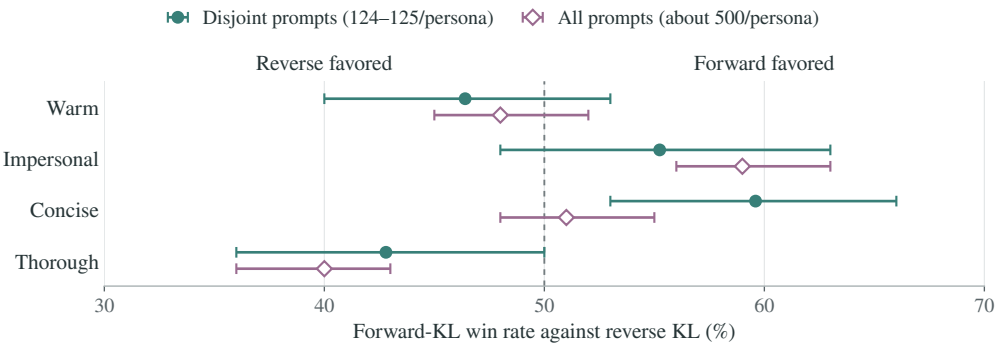

Figure 5: **KL effects depend on persona and request assignment.** Forward-KL win rates against reverse KL; horizontal lines are reported 95% bootstrap intervals. Values above 50% favor forward KL. Filled circles use disjoint request groups; open diamonds use every test task for each persona. The two evaluations overlap and are not independent replications.

H.2 Teacher and response diagnostics

The diagnostic analysis reconstructs the training teacher: the frozen base conditioned on the persona card and the Qwen3.5-27B critic's feedback on each answer. It compares teacher and head distributions on both heads' answers from the disjoint request groups. Reverse-KL heads have entropy close to the teacher's and consistently below the forward-KL heads, while top-token agreement changes little (Table 22). Forward KL also retains more of the teacher's probability mass on hedge words, most clearly for impersonal and concise heads. These are distributional diagnostics on observed prefixes, not measurements of diversity across sampled responses.

Table 22: **Diagnostics under the reconstructed training teacher.** T, F, and R denote teacher, forward-KL head, and reverse-KL head. Top-token agreement compares the head's most probable token with the teacher's. Hedge-mass retention is the share of the teacher's hedge-word probability retained by the head. Values are the reported averages.

	Token entropy			Top-token agreement		Hedge-mass retention	
Persona	T	F	R	F	R	F	R
Warm	0.78	1.08	0.82	0.79	0.80	0.91	0.90
Impersonal	0.61	0.98	0.64	0.80	0.80	0.81	0.76
Concise	0.32	0.93	0.31	0.73	0.73	0.57	0.45
Thorough	0.81	1.00	0.83	0.88	0.88	0.94	0.93

Response-level summaries connect these differences to behavior. Forward-KL impersonal answers contain more hedges (0.85 versus 0.70 per 100 words), and the judge selects the more hedged answer in 37/58 eligible decisive comparisons. Reverse-KL thorough answers are longer (median 1,257 versus 1,140 words); the judge selects the longer answer in 72/91 such comparisons. These associations are consistent with preferences for qualified wording and greater detail. Forward-KL concise answers are slightly shorter (median 8 versus 9 words), and shorter answers win 64/70 eligible decisive comparisons, but the preference advantage itself is sensitive to request assignment. Reverse-KL warm answers less often express their own opinion (66 versus 119 instances in roughly 500 responses), without a clear aggregate preference advantage.

These diagnostics support an association between probability allocation and persona expression, rather than proving that preserving or suppressing particular tokens causes the observed preferences. Length and hedging are proxies that can also influence the judge directly. The two losses use different retained token supports, and reverse KL approximates teacher probabilities outside its cache. Each objective has one training run; bootstrap intervals reflect prompt variation, not training-seed variation. The expanded evaluation is a check on request assignment using the same test tasks, and this ablation does not evaluate tutoring or establish independently measured task-quality gains.