LEARNING WHAT TO RECOMMEND AND CACHE: COST-GUIDED MULTI-AGENT CONTROL AT THE EDGE

Anonymous authors
Paper under double-blind review

ABSTRACT

Recommendation-enabled edge caching goes beyond anticipating demand because it can reshape user preferences and change what users request. Recommendation response varies across users, but an edge controller does not directly observe the personality traits that may characterize this heterogeneity. To infer these latent traits from observable rating behavior, we develop Rating-derived Indicators for Trait Estimation (RITE), a predictor based on an incremental Broad Learning System (BLS). RITE extracts six behavioral descriptors from each user's rating history and uses them to predict the user's Big Five personality traits. We then combine the predicted personality traits with recommendation-list features to estimate each user's response to recommendations. Building on this representation, we formulate joint recommendation and caching as a sequential multi-agent control problem and introduce Demand Reshaping via Action Projection at the Edge (DRAPE). DRAPE conditions decentralized actors and centralized critics on the inferred user model, while a cost-guided projection converts continuous scores into capacity-feasible caching and top-K recommendation decisions. Experimental results show that DRAPE consistently achieves lower total cost than five representative baselines, while ablation studies demonstrate the effectiveness of its cost-guided action projection.

1 INTRODUCTION

Edge caching places popular content at small base stations (SBSs) near mobile users, while device-to-device (D2D) delivery extends local service to content stored on nearby devices. Requests for locally available content can bypass the core network, reducing repeated backhaul retrieval and content-delivery latency (Luo et al., 2025; Sun et al., 2026; Jiang et al., 2025; Deng et al., 2025; Jiang et al., 2026; Nguyen et al., 2026). Effective offloading, however, requires cached content to match incoming requests despite limited storage, user mobility, intermittent contacts, and evolving interests. Recommendation policies can further improve request–cache alignment by prioritizing content available at an SBS or a D2D helper.

Recommendation-enabled caching has shown that network demand can be shaped rather than merely predicted. By coordinating recommended and cached content, existing methods create additional opportunities for cache hits and D2D delivery while reducing backhaul traffic (Liu et al., 2025; Sun et al., 2026; Cai et al., 2025; Zhang et al., 2025b). These methods, however, generally represent each user through ratings, request histories, or a fixed preference distribution. Such representations describe what users have preferred, but they cannot capture response differences that affect how recommendations reshape the demand presented to the caching policy.

User personality may explain recommendation-response differences that ratings and request histories alone leave unresolved (Chen et al., 2026a; Zhang et al., 2025a; Ma et al., 2025). Existing recommendation-aware caching models represent personality through a psychological threshold or acceptance probability (Zhou et al., 2025b; Zhang et al., 2025a), but a single scalar cannot describe how a multidimensional profile affects the response to a particular recommendation list. Detailed personality labels are privacy-sensitive and difficult to collect at scale (Song et al., 2025; Li et al., 2026; Han et al., 2025; Zhao et al., 2025b), whereas recommender systems routinely record rating histories that offer a less intrusive basis for inference even when the observations are sparse. However, an inferred personality representation does not fully characterize recommendation response, which

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.

also depends on the contents included in the list. Recommendation and caching decisions must also be coordinated across SBSs because overlapping coverage and time-varying D2D contacts allow local actions to affect common users and delivery paths (Liang et al., 2026; Sun et al., 2026; Yang et al., 2025; Xu et al., 2025; Chen et al., 2026b;c).

According to the above analysis, there are three challenges that need to be addressed: (i) how to accurately infer a multidimensional personality representation from sparse ratings without collecting detailed personality labels from every edge user at decision time; (ii) how to relate multidimensional personality and recommendation-list context to user-specific response and the corresponding changes in edge demand; and (iii) how to coordinate sequential recommendation and caching across locally informed SBSs whose decisions are coupled by overlapping coverage and time-varying D2D contacts while satisfying storage and top-K constraints. To address these challenges, we first develop Rating-derived Indicators for Trait Estimation (RITE) to infer multidimensional personality traits from sparse ratings. Then, we model how personality and recommendation-list context jointly shape user responses and the resulting demand. We further develop Demand Reshaping via Action Projection at the Edge (DRAPE) to coordinate recommendation and caching across SBSs. The contributions are summarized as follows:

- RITE is developed as a behavior-to-personality prediction method based on an incremental BLS. Six behavioral descriptors are extracted from each rating history and mapped to a Big Five representation, which is then combined with recommendation-list features to estimate user-specific recommendation response.
- Joint recommendation and caching are formulated as a constrained sequential decision problem, and DRAPE is developed as a behavior-conditioned multi-agent controller based on Reinforcement Learning (RL). Decisions across SBSs are coordinated by decentralized actors and centralized critics, while continuous actor scores are converted into capacity-feasible cache placements and top-K recommendation lists through cost-guided projection.
- A trace-driven evaluation is conducted under a unified protocol with five representative baselines. Across ten paired seeds, DRAPE is shown to achieve lower total cost than every comparator. Component behavior and robustness are further examined through prediction checks, ablations, rebuilt user mappings, operating-condition tests, and scalability measurements.

2 Related Work

Learning-based methods have been developed to jointly optimize recommendation and edge caching, with D2D delivery incorporated into several formulations (Zhou et al., 2024; Wu et al., 2024). Robust and preference-aware formulations consider popularity uncertainty, edge failures, user preferences, and mobility (Luo et al., 2025; Zhou et al., 2025b; Jiang et al., 2025; Dai et al., 2025; Huang et al., 2026; Qiu et al., 2025), whereas multi-agent methods coordinate cache placement among distributed edge nodes (Liang et al., 2026; Chen et al., 2026b;c;d). D2D-enhanced designs extend joint placement from edge servers to user devices (Sun et al., 2026; Hu et al., 2025; Deng et al., 2025; Nguyen et al., 2026), and recent recommendation models distinguish natural preference from recommendation-induced decisions (Liu et al., 2025; Cai et al., 2025). However, user response is still represented mainly by ratings, request histories, or fixed preference estimates. Such representations do not explicitly describe how users with different personality profiles respond to a particular recommendation list.

Personality-aware recommendation follows a different line of inquiry, shifting attention from what users prefer to how users differ. Some methods integrate profiles, preferences, and personality-related characteristics into a unified user representation (Zhang et al., 2025a; Shi et al., 2025; Kim et al., 2025; Wang et al., 2025), whereas personality-identification methods infer traits from behavioral traces when labeled data are limited (Song et al., 2025; Ma et al., 2025). Human-grounded studies further associate Big Five traits with heterogeneous responses to common content (Yang et al., 2026; Zhang et al., 2025b), while privacy research shows that learned user representations may retain sensitive attributes (Li et al., 2026; Han et al., 2025; Zhao et al., 2025b). However, these studies focus on personality inference or recommendation response outside edge caching. They do not connect rating-derived personality with list-specific request changes or joint recommendation and caching under storage and D2D constraints.

3 PROBLEM FORMULATION

Let $\mathcal{U}$, $\mathcal{M}$, $\mathcal{F}$, and $\mathcal{T} = \{1, \ldots, T\}$ denote the sets of users, SBSs, contents, and time slots, respectively. Each content $f \in \mathcal{F}$ has size s_f, and each SBS $m \in \mathcal{M}$ has cache capacity C_m^t at slot $t \in \mathcal{T}$. For each user $u \in \mathcal{U}$, $\mathcal{M}_u \subseteq \mathcal{M}$ denotes the set of SBSs accessible to that user, where $|\mathcal{M}_u| > 1$ indicates overlapping coverage.

Users may obtain a requested content through D2D transmission. Let $h_{i,f}^t \in \{0, 1\}$ indicate whether potential helper i holds content f, and let $p_{i,u}^{\text{con}}$ denote the probability that i contacts u within the tolerable delay. Under conditional independence among potential helpers, the probability that content f is available to user u through D2D transmission in slot t is expressed as Eq. (1):

$$D_{u,f}^t = 1 - \prod_{i \in \mathcal{U} \setminus \{u\}} \left(1 - h_{i,f}^t p_{i,u}^{\text{con}}\right). \tag{1}$$

In practice, each user has latent personal characteristics, but detailed personality labels are difficult to obtain at the edge because of privacy concerns. Let $\mathbf{p}_u \in [0, 1]^{d_p}$ represent the latent personality profile of user $u \in \mathcal{U}$. The edge controller observes only a sparse rating history $\mathcal{R}_u = \{(f, r_{u,f}) : f \in \mathcal{F}_u\}$, where $\mathcal{F}_u \subseteq \mathcal{F}$ is the set of contents rated by user u and $r_{u,f} \in [r^{\min}, r^{\max}]$. Thus, $\mathbf{p}_u$ and its relation to $\mathcal{R}_u$ are unknown when the optimization problem is formulated.

To characterize user u's preference before recommendation, let $\widetilde{r}_{u,f} \in [r^{\min}, r^{\max}]$ denote the baseline score of content f. The score equals $r_{u,f}$ when a rating is observed and otherwise represents the user's unobserved preference. The corresponding normalized preference is defined as Eq. (2):

$$q_{u,f} = \frac{\widetilde{r}_{u,f} - r^{\min}}{r^{\max} - r^{\min}} \in [0, 1]. \tag{2}$$

Let $y_{u,f}^t$ denote the recommendation decision for content f and user u in slot t, where $y_{u,f}^t = 1$ indicates that f is recommended to u and $y_{u,f}^t = 0$ otherwise. With $\mathbf{y}_u^t = [y_{u,f}^t]_{f \in \mathcal{F}}$ collecting these decisions, $\mathbf{c}_u^t$ summarizes the corresponding recommendation list. Then, $r_u^{\text{res},t} \in [0, 1]$ characterizes how strongly user u responds to the list. To account for differences across contents, $\xi_{u,f}$ modulates this response according to the baseline preference and is defined as Eq. (3):

$$\xi_{u,f} = 4 q_{u,f}(1 - q_{u,f}). \tag{3}$$

Although $r_u^{\text{res},t}$ depends on user personality and recommendation-list context, the edge controller cannot observe its value directly. For $\xi_{u,f}$, multiplying $q_{u,f}(1 - q_{u,f})$ by 4 normalizes the term to $[0, 1]$ because its maximum value is $1/4$ at $q_{u,f} = 0.5$. Hence, $\xi_{u,f}$ is largest for moderate baseline preference and becomes zero at both endpoints. Incorporating the response coefficient and content-level influence into the baseline preference yields the recommendation-conditioned demand weight as Eq. (4):

$$\rho_{u,f}^t = q_{u,f} + y_{u,f}^t r_u^{\text{res},t} \xi_{u,f}(1 - q_{u,f}). \tag{4}$$

When $y_{u,f}^t = 0$, the demand weight remains $q_{u,f}$. A recommended content receives an increment determined by the user's response, the content-level influence, and the remaining preference range. To evaluate how this demand is served, let α_m, α_e, and α_u denote the per-unit cellular, backhaul, and D2D costs. Caching is represented by $x_{m,f}^t$, where $x_{m,f}^t = 1$ indicates that SBS m caches content f in slot t and $x_{m,f}^t = 0$ otherwise. The minimum cellular cost for delivering content f to user u is defined as Eq. (5):

$$c_{u,f}^{\text{cell},t} = \min_{m \in \mathcal{M}_u} s_f \left[\alpha_m + (1 - x_{m,f}^t)\alpha_e\right]. \tag{5}$$

When D2D delivery is attempted, failure leads to cellular fallback. The resulting service cost is defined as Eq. (6):

$$c_{u,f}^{\text{ser},t} = \min\left\{c_{u,f}^{\text{cell},t}, D_{u,f}^t s_f \alpha_u + (1 - D_{u,f}^t) c_{u,f}^{\text{cell},t}\right\}. \tag{6}$$

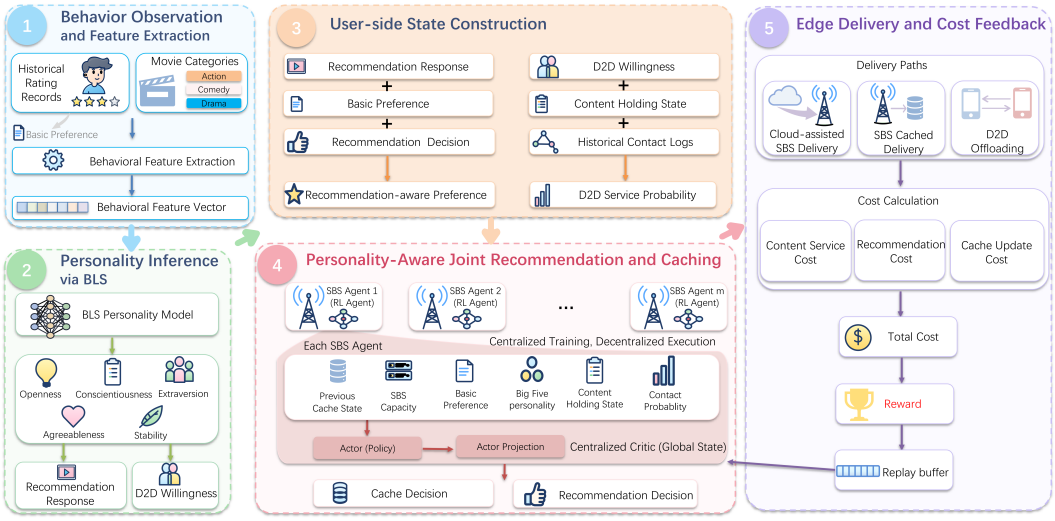

Figure 1: Overall framework of RITE and DRAPE. RITE infers user personality from rating behavior, while DRAPE coordinates recommendation and caching using the resulting user-side state and edge-delivery feedback.

Let $\mathbf{X} = \{x^t_{m,f}\}$ and $\mathbf{Y} = \{y^t_{u,f}\}$ collect all caching and recommendation decisions. Then, we formulate the objective of minimizing the total system cost as Eq. (7):

$$
\begin{aligned}
\min_{\mathbf{X},\mathbf{Y}} \quad & C^{\text{total}} = \sum_{t\in\mathcal{T}}\sum_{u\in\mathcal{U}}\sum_{f\in\mathcal{F}} \rho^t_{u,f} c^{\text{ser},t}_{u,f} + \lambda_r \sum_{t\in\mathcal{T}}\sum_{u\in\mathcal{U}}\sum_{f\in\mathcal{F}} y^t_{u,f}(1-q_{u,f}) \\
& \qquad + \lambda_c \sum_{t=2}^{T}\sum_{m\in\mathcal{M}}\sum_{f\in\mathcal{F}} \left[x^t_{m,f} - x^{t-1}_{m,f}\right]^+ \\
\text{s.t.} \quad & \text{(C1)} \quad \sum_{f\in\mathcal{F}} s_f x^t_{m,f} \le C^t_m, && \forall m\in\mathcal{M},\ t\in\mathcal{T}, \\
& \text{(C2)} \quad \sum_{f\in\mathcal{F}} y^t_{u,f} = K, && \forall u\in\mathcal{U},\ t\in\mathcal{T}, \\
& \text{(C3)} \quad x^t_{m,f}, y^t_{u,f} \in \{0,1\}, && \forall^{m\in\mathcal{M},\ u\in\mathcal{U},}_{f\in\mathcal{F},\ t\in\mathcal{T}},
\end{aligned}
\tag{7}
$$

where $\lambda_r, \lambda_c \ge 0$ are weighting coefficients and $[z]^+ = \max\{z, 0\}$. Constraint (C1) enforces the cache capacity of each SBS, (C2) selects exactly K contents for each user, and (C3) restricts both decisions to binary values.

4 Problem Solving

In this section, we first introduce RITE to infer users' Big Five personality traits from rating histories. Then, a multi-agent RL method named DRAPE is proposed to jointly optimize recommendation and caching decisions, as illustrated in figure 1.

4.1 RITE for Personality Prediction

To address the difficulty of inferring latent user characteristics from sparse rating histories, we introduce the Big Five traits as a multidimensional personality representation and propose RITE, a behavior-to-personality prediction method. RITE partitions contents into K_p popularity strata $\mathcal{K} = \{1, \ldots, K_p\}$ using reference-user rating counts, where K_p is configurable, and lets $\kappa(f) \in \mathcal{K}$ denote the stratum containing content f. For user u, the rated-content distribution over these strata is

calculated as Eq. (8):

$$p_{u,k} = \frac{1}{|\mathcal{F}_u|} \sum_{f \in \mathcal{F}_u} \mathbf{1}\{\kappa(f) = k\}, \tag{8}$$

where $\mathbf{1}\{\cdot\}$ is the indicator function and $\{p_{u,k}\}_{k \in \mathcal{K}}$ is the popularity profile of user u. Together with the observed rating values and history length, this profile produces six behavioral descriptors as Eq. (9):

$$\begin{aligned} \mu_u &= \frac{1}{|\mathcal{F}_u|} \sum_{f \in \mathcal{F}_u} r_{u,f}, & \sigma_u^2 &= \frac{1}{|\mathcal{F}_u|} \sum_{f \in \mathcal{F}_u} (r_{u,f} - \mu_u)^2, \\ H_u &= -\frac{1}{\log |\mathcal{K}|} \sum_{\substack{k \in \mathcal{K} \\ p_{u,k} > 0}} p_{u,k} \log p_{u,k}, & B_u &= \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} \mathbf{1}\{p_{u,k} > 0\}, \\ A_u &= \log(1 + |\mathcal{F}_u|), & E_u &= \frac{1}{|\mathcal{F}_u|} \sum_{f \in \mathcal{F}_u} \mathbf{1}\{r_{u,f} \le r_l \text{ or } r_{u,f} \ge r_h\}. \end{aligned} \tag{9}$$

The fixed thresholds r_l and r_h satisfy $r^{\min} \le r_l < r_h \le r^{\max}$. The tuple (μ_u, σ_u^2, E_u) characterizes rating style, (H_u, B_u) measures content-selection diversity, and A_u records rating activity. The factor $1/\log|\mathcal{K}|$ normalizes H_u to $[0, 1]$, while the logarithm in A_u limits the influence of highly active users. Their ordered vector $\mathbf{e}_u = [\mu_u, \sigma_u^2, H_u, B_u, A_u, E_u]$ provides the input to the subsequent personality mapping.

Let $\mathcal{U}_{\text{ref}}$ index reference users with rating-derived representations and Big Five vectors $(\mathbf{e}_u, \mathbf{p}_u)$, and let $\mathcal{U}_{\text{tar}}$ index target users with only $\mathbf{e}_u$. The reference matrices are $\mathbf{E}_{\text{ref}} = [\mathbf{e}_u^\top]_{u \in \mathcal{U}_{\text{ref}}} \in \mathbb{R}^{n_{\text{ref}} \times 6}$ and $\mathbf{P}_{\text{ref}} = [\mathbf{p}_u^\top]_{u \in \mathcal{U}_{\text{ref}}} \in [0, 1]^{n_{\text{ref}} \times 5}$, where $n_{\text{ref}} = |\mathcal{U}_{\text{ref}}|$. From the column-wise mean $\boldsymbol{\mu}_{\text{ref}}$ and standard deviation $\boldsymbol{\sigma}_{\text{ref}}$ of $\mathbf{E}_{\text{ref}}$, RITE obtains $\overline{\mathbf{e}}_u = (\mathbf{e}_u - \boldsymbol{\mu}_{\text{ref}}) \oslash \boldsymbol{\sigma}_{\text{ref}}$ for every reference and target user and forms $\overline{\mathbf{E}}_{\text{ref}} = [\overline{\mathbf{e}}_u^\top]_{u \in \mathcal{U}_{\text{ref}}}$, replacing near-zero entries of $\boldsymbol{\sigma}_{\text{ref}}$ by one. Following the flat width-expansion architecture in Chen & Liu (2018), the standardized reference matrix enters a fixed feature map and produces d_z feature nodes as Eq. (10):

$$\mathbf{Z}_{\text{ref}} = \tanh([\overline{\mathbf{E}}_{\text{ref}}, \mathbf{1}]\mathbf{W}_e), \tag{10}$$

where $\mathbf{1}$ is an all-ones column and $\mathbf{W}_e \in \mathbb{R}^{7 \times d_z}$ is sampled once and fixed. The resulting feature nodes define the initial design matrix $\mathbf{A}_{\text{ref}}^{(0)} = [\overline{\mathbf{E}}_{\text{ref}} \mid \mathbf{Z}_{\text{ref}}]$. At expansion round ℓ, the fixed feature nodes enter the ℓth enhancement block and produce d_h additional nodes as Eq. (11):

$$\mathbf{H}_{\text{ref}}^{(\ell)} = \tanh([\mathbf{Z}_{\text{ref}}, \mathbf{1}]\mathbf{W}_h^{(\ell)}). \tag{11}$$

Here, $\mathbf{W}_h^{(\ell)} \in \mathbb{R}^{(d_z+1) \times d_h}$ is the ℓth d_h-column block of a complete matrix $\mathbf{W}_h \in \mathbb{R}^{(d_z+1) \times L d_h}$ sampled once and fixed. Appending this block to the current design matrix produces the expanded representation as Eq. (12):

$$\mathbf{A}_{\text{ref}}^{(\ell)} = [\mathbf{A}_{\text{ref}}^{(\ell-1)} \mid \mathbf{H}_{\text{ref}}^{(\ell)}], \tag{12}$$

where $\mid$ denotes column-wise concatenation and $\ell = 1, \ldots, L$. The expanded design matrix and reference personality vectors then determine the ridge readout at width $\ell \in \{0, \ldots, L\}$ as Eq. (13):

$$\mathbf{W}_{\text{out}}^{(\ell)} = \left((\mathbf{A}_{\text{ref}}^{(\ell)})^\top \mathbf{A}_{\text{ref}}^{(\ell)} + \lambda \mathbf{I} \right)^{-1} (\mathbf{A}_{\text{ref}}^{(\ell)})^\top \mathbf{P}_{\text{ref}}, \tag{13}$$

in which $\lambda > 0$ is the ridge coefficient and $\mathbf{I}$ is an identity matrix of the required size. After each expansion, the implementation obtains the equivalent readout from the previous inverse Gram matrix through a Schur-complement block update. For target user u, the fixed maps in Eqs. (10) and (11) give $\mathbf{z}_u = \tanh([\overline{\mathbf{e}}_u^\top, 1]\mathbf{W}_e)$ and $\mathbf{h}_u^{(\ell)} = \tanh([\mathbf{z}_u, 1]\mathbf{W}_h^{(\ell)})$. Concatenating these quantities with the standardized behavioral representation and applying the final readout produces the target personality vector as Eq. (14):

$$\widehat{\mathbf{p}}_u = \text{clip}_{[0,1]^5} \left([\overline{\mathbf{e}}_u^\top \mid \mathbf{z}_u \mid \mathbf{h}_u^{(1)} \mid \cdots \mid \mathbf{h}_u^{(L)}] \mathbf{W}_{\text{out}}^{(L)} \right), \tag{14}$$

where $\text{clip}_{[0,1]^5}(\cdot)$ bounds every predicted Big Five component to $[0, 1]$. The inferred personality vector is paired with the recorded recommendation-list context because personality alone does not

Algorithm 1 RITE for Personality Prediction

Require: Reference rating histories with Big Five labels, target rating histories, $K_p, d_z, d_h, L, \lambda, r_l$, and r_h

Ensure: Target personality vectors $\{\widehat{\mathbf{p}}_u : u \in \mathcal{U}_{\mathrm{tar}}\}$

1: Build K_p strata from reference-user item counts and fix κ
2: Compute every reference and target descriptor $\mathbf{e}_u$ using Eqs. (8)–(9)
3: Estimate $\boldsymbol{\mu}_{\mathrm{ref}}$ and $\boldsymbol{\sigma}_{\mathrm{ref}}$ from reference descriptors only
4: Replace near-zero scales by one and standardize both user sets with the reference statistics
5: Sample once and fix $\mathbf{W}_e$ and the complete block matrix $\mathbf{W}_h$
6: Compute $\mathbf{Z}_{\mathrm{ref}}$ and form $\mathbf{A}_{\mathrm{ref}}^{(0)}$ using Eq. (10)
7: Solve the width-zero ridge problem in Eq. (13) and retain its inverse Gram matrix
8: **for** $\ell = 1, \ldots, L$ **do**
9: Generate $\mathbf{H}_{\mathrm{ref}}^{(\ell)}$ with the fixed block $\mathbf{W}_h^{(\ell)}$ using Eq. (11)
10: Append the block to obtain $\mathbf{A}_{\mathrm{ref}}^{(\ell)}$ using Eq. (12)
11: Update the inverse Gram matrix and $\mathbf{W}_{\mathrm{out}}^{(\ell)}$ through the equivalent Schur-complement block update
12: **end for**
13: Propagate the standardized target descriptors through the same fixed maps in the training block order
14: Compute and clip the target predictions using Eq. (14)
15: **return** $\{\widehat{\mathbf{p}}_u : u \in \mathcal{U}_{\mathrm{tar}}\}$

determine a user's response. Let $\bar{\mathbf{c}}_u$ denote the context features for user u. A cross-fitted ridge estimator produces the policy-visible response as Eq. (15):

$$\widehat{r}_u^{\mathrm{res}} = \mathrm{clip}_{[0,1]}\left(\widehat{g}_\theta(\bar{\mathbf{c}}_u, \widehat{\mathbf{p}}_u)\right). \tag{15}$$

The function $\widehat{g}_\theta$ maps list context and RITE-inferred personality to a response score. Out-of-fold fitting prevents user u's response label from contributing to its own estimate. Because the available response record is user-level rather than slot-specific, $\widehat{r}_u^{\mathrm{res}}$ approximates $r_u^{\mathrm{res},t}$ across slots and is supplied to DRAPE.

Algorithm 1 shows the details of the proposed RITE. We first construct the behavioral representation (lines 1–2), performs reference-only normalization (lines 3–4), initializes the fixed BLS maps and ridge readout (lines 5–7), incrementally expands the design through Schur-complement updates (lines 8–12), and applies the fixed mapping to target users (lines 13–15).

4.2 DRAPE for Joint Recommendation and Caching

Appendix A proves that Eq. (7) is NP-hard by reduction from 0–1 knapsack. DRAPE therefore seeks low-cost feasible decisions through CTDE and cost-guided projection (Hung et al., 2025; Zhou et al., 2025a; Holder et al., 2025; Qiu et al., 2025).

Let $M = |\mathcal{M}|$ denote the number of SBS agents. Agent m observes the users in $\mathcal{U}_m$ and forms its local observation as Eq. (16):

$$\mathbf{o}_m^t = [\mathbf{x}_m^{t-1}, C_m^t, \mathbf{q}_m, \widehat{\mathbf{p}}_m, \mathbf{h}^t, \mathbf{g}_m^t]. \tag{16}$$

The vector $\mathbf{q}_m$ collects local baseline preferences, $\widehat{\mathbf{p}}_m$ collects the predicted Big Five representations, $\mathbf{h}^t$ records content holdings, and $\mathbf{g}_m^t$ contains the contact information needed for Eq. (1). Centralized training uses the concatenated state $\mathbf{s}^t = [\mathbf{o}_1^t, \ldots, \mathbf{o}_M^t]$. Actor m maps its local observation to continuous caching and recommendation scores as Eq. (17):

$$\widetilde{\mathbf{a}}_m^t = \mu_m(\mathbf{o}_m^t; \theta_m^\mu) = [\widetilde{\mathbf{x}}_m^t, \widetilde{\mathbf{y}}_m^t]. \tag{17}$$

All agents share $R^t = -C_t^{\mathrm{total}}$. Following MADDPG (Lowe et al., 2017; Yang et al., 2025; Xu et al., 2025; Chen et al., 2026b;c), critic Q_m uses the global state and continuous joint action during training, whereas actor μ_m uses only $\mathbf{o}_m^t$ during execution. Replay transitions retain the pre-projection action and terminal flag d^j, yielding the Bellman target as Eq. (18):

$$z_m^j = R^j + \gamma(1 - d^j) Q'_m\left(\mathbf{s}^{j+1}, \mu'_1(\mathbf{o}_1^{j+1}), \ldots, \mu'_M(\mathbf{o}_M^{j+1})\right), \tag{18}$$

Algorithm 2 DRAPE for Joint Recommendation and Caching

Require: RITE-inferred traits, response estimates, local observations, candidate sets $\mathcal{C}^t$ with $|\mathcal{C}^t| \geq K$, γ, and τ
Ensure: Feasible binary caching and recommendation decisions $(\mathbf{X}^t, \mathbf{Y}^t)$
1: Initialize actors, centralized critics, target networks, and replay buffer $\mathcal{D}$
2: **function** PROJECT($\widetilde{\mathbf{a}}^t$)
3: Set $\mathbf{X}^t = \mathbf{0}$ and, for each SBS, rank $\mathcal{C}^t$ by $\Omega^t_{m,f}$ and admit contents while capacity remains
4: Set $\mathbf{Y}^t = \mathbf{0}$ and, for each user, rank $\mathcal{C}^t$ by $\Pi^t_{u,f}$ and select the top K contents
5: **return** $(\mathbf{X}^t, \mathbf{Y}^t)$
6: **end function**
7: **for** each training episode and slot t **do**
8: Generate noisy actor scores, project and execute them, and store the transition in $\mathcal{D}$
9: When a minibatch is available, update the critics, actors, and targets using Eqs. (18)–(21)
10: **end for**
11: **return** PROJECT($\{\mu_m(\mathbf{o}^t_m; \theta^\mu_m)\}_{m=1}^M$) at execution

with discount factor γ and minibatch $\mathcal{B}$, the centralized critic minimizes Eq. (19):

$$L^Q_m(\theta^Q_m) = \frac{1}{|\mathcal{B}|} \sum_{j \in \mathcal{B}} \left[Q_m(\mathbf{s}^j, \widetilde{\mathbf{a}}^j_1, \ldots, \widetilde{\mathbf{a}}^j_M; \theta^Q_m) - z^j_m \right]^2. \tag{19}$$

The resulting action derivative updates actor m through Eq. (20):

$$\nabla_{\theta^\mu_m} J_m = \frac{1}{|\mathcal{B}|} \sum_{j \in \mathcal{B}} \left(\frac{\partial \mu_m(\mathbf{o}^j_m; \theta^\mu_m)}{\partial \theta^\mu_m} \right)^\top \nabla_{\widetilde{\mathbf{a}}_m} Q_m \left(\mathbf{s}^j, \mathbf{a}^{\mu,j}_1, \ldots, \mathbf{a}^{\mu,j}_M; \theta^Q_m \right), \tag{20}$$

where $\mathbf{a}^{\mu,j}_m = \mu_m(\mathbf{o}^j_m; \theta^\mu_m)$ and $\mathbf{a}^{\mu,j}_n = \mathrm{sg}[\mu_n(\mathbf{o}^j_n; \theta^\mu_n)]$ for $n \neq m$, where sg stops the gradient. Target parameters follow the Polyak update in Eq. (21):

$$\vartheta'_m \leftarrow \tau \vartheta_m + (1 - \tau) \vartheta'_m, \qquad \vartheta_m \in \{\theta^\mu_m, \theta^Q_m\}, \tag{21}$$

in which the rate $\tau \in (0, 1]$ controls target-network tracking. Training operates on continuous pre-projection actions, whereas decentralized execution retains each local actor and the nongradient projection. The recommendation-conditioned demand used by this projection is given by Eq. (22):

$$\bar{\rho}^t_{u,f} = q_{u,f} + \widehat{r}^{\,\mathrm{res}}_u \xi_{u,f} (1 - q_{u,f}). \tag{22}$$

Multiplying this demand weight by the probability that the content is unavailable through D2D gives the cache-benefit estimate for content f at SBS m as Eq. (23):

$$B^t_{m,f} = \sum_{u \in \mathcal{U}_m} \bar{\rho}^t_{u,f} (1 - D^t_{u,f}) s_f \alpha_e. \tag{23}$$

Normalizing this estimate by content size yields a cache priority that combines the actor score with a marginal-cost proxy as Eq. (24):

$$\Omega^t_{m,f} = \widetilde{x}^t_{m,f} + \kappa_x \frac{B^t_{m,f}}{s_f + \epsilon}. \tag{24}$$

The coefficient κ_x controls the cache guidance and $\epsilon > 0$ stabilizes size normalization. Each SBS scans contents in descending $\Omega^t_{m,f}$ order and caches every item that fits in the remaining capacity, producing a cache decision that satisfies constraint (C1). After caching is fixed, recommending content f produces the demand increment as Eq. (25):

$$\Delta \rho^t_{u,f} = \widehat{r}^{\,\mathrm{res}}_u \xi_{u,f} (1 - q_{u,f}). \tag{25}$$

An uncached request served through the cloud incurs $c^{\mathrm{cloud},t}_{u,f} = s_f(\alpha_m + \alpha_e)$. Relative to the current service cost, $G^t_{u,f} = c^{\mathrm{cloud},t}_{u,f} - c^{\mathrm{ser},t}_{u,f}$ gives the saving associated with each recommendation-induced

Table 1: Prediction performance on the 78 held-out mapped users. All targets are normalized to $[0, 1]$, and response estimates are obtained out of fold. Lower MAE and RMSE and higher Pearson r are better. O, C, E, A, and S denote openness, conscientiousness, extraversion, agreeableness, and emotional stability, respectively.

Personality prediction

Metric	Predictor	O	C	E	A	S	Macro
MAE ↓	RITE	0.1334	0.2131	0.2023	0.1743	0.2136	0.1873
	Training mean	0.1342	0.2190	0.2104	0.1745	0.2060	0.1888
RMSE ↓	RITE	0.1610	0.2547	0.2479	0.2133	0.2477	0.2276
	Training mean	0.1610	0.2575	0.2546	0.2107	0.2401	0.2276
Pearson r ↑	RITE	0.143	0.153	0.234	0.088	−0.080	0.108
	Training mean	–	–	–	–	–	–

Recommendation-response prediction

Predictor	MAE ↓	RMSE ↓	Pearson r ↑
List context + RITE traits	0.1373	0.1612	0.008
List context only	0.1359	0.1591	0.044

request. DRAPE incorporates this saving and the deviation from baseline preference into the actor score as Eq. (26):

$$\Pi^t_{u,f} = \max_{m \in \mathcal{M}_u} \left\{ \widetilde{y}^t_{m,u,f} + \kappa_y \left[\Delta\rho^t_{u,f} G^t_{u,f} - \lambda_r (1 - q_{u,f}) \right] \right\}. \tag{26}$$

Here, κ_y controls the strength of cost guidance, which raises the actor score when the induced requests can be served more cheaply and penalizes contents that depart from baseline preference. For users covered by multiple SBSs, the maximum retains the highest local score. Selecting the top K candidates produces the binary recommendation decision and satisfies constraint (C2). Candidate contents are drawn from $\mathcal{C}^t \subseteq \mathcal{F}$, which is shared by all methods and satisfies $|\mathcal{C}^t| \geq K$.

Algorithm 2 initializes the trainable components (line 1), defines the feasible projection (lines 2–6), alternates interaction and learning (lines 7–10), and returns the deployed decisions (line 11).

For an $n_b \times d_b$ expanded design matrix, RITE fitting costs $O(n_b d_b^2 + d_b^3)$ and inference costs $O(d_b)$ per user. With $c = |\mathcal{C}^t|$, DRAPE projection costs $O(|E_t|c + \sum_m n_m c + Mc \log c + \sum_u |\mathcal{M}_u|c + |\mathcal{U}|c \log K)$. Actor execution is $O(MP_\mu)$, and centralized critics add $O(MP_Q)$ per update, up to batch factors.

5 Experimental Evaluation

Data and protocol. We pair 78 held-out Personality users with the 78 INFOCOM users and train RITE on the remaining 1,742 users. The simulator uses 100 contents, five SBSs with 15% cache capacity, five-item device caches, $K = 3$, 120-second slots, and separate training and test traces. Each method trains for 120 episodes and is evaluated for 30 episodes across ten matched seeds (2027–2036). Hyperparameters use training data only, and paired Wilcoxon tests apply Holm correction. Appendices C and D provide the complete construction and settings.

Prediction checks. Table 1 reports both prediction tasks on the held-out mapped users. RITE yields a lower macro MAE than the training-mean predictor, although the gain varies across traits and macro RMSE is unchanged. Adding RITE traits does not improve response prediction, so they are treated as noisy side information (Appendix F).

Scenario-adapted comparators. DRAPE is compared with EC-MADRL, OOA, DRA-SAC, CO-HCTO, and uEDC-L (Liang et al., 2026; Zhou et al., 2025b; Dong et al., 2025; Zhao et al., 2025a; Luo et al., 2025). All methods share data, behavioral inputs, SBS assignments, candidates, initial states, stochastic inputs, constraints, and training budget, but retain their own policies and feasibility decoders. The scenario adaptations are not author-code reproductions and are detailed in Appendix D.

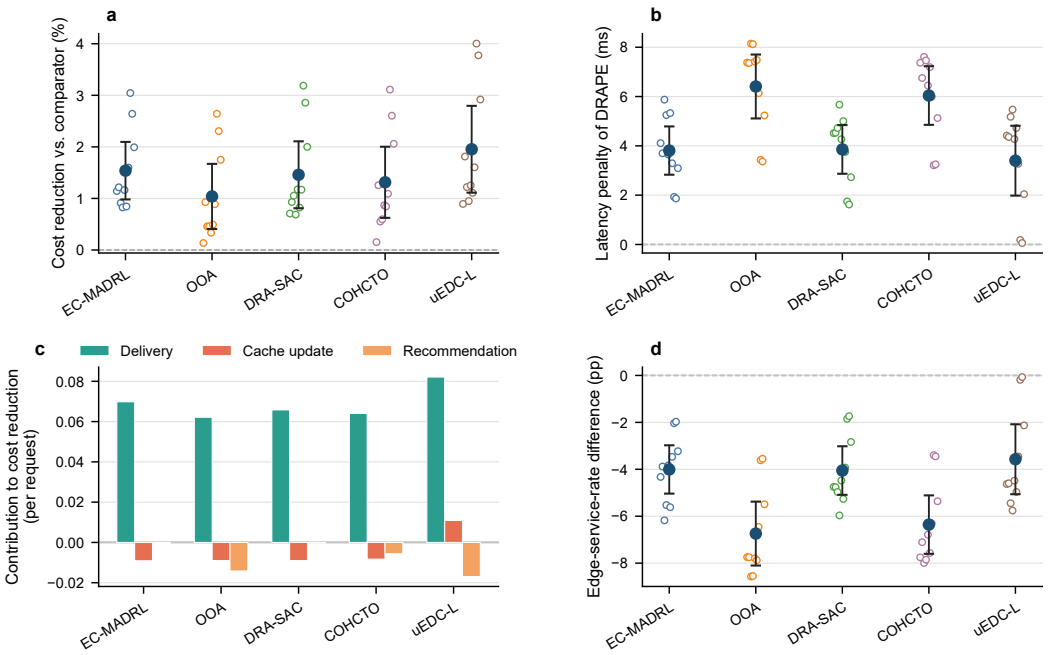

Figure 2: Ten-seed paired comparison. Panels show (a) cost reduction, (b) latency difference, (c) cost decomposition, and (d) edge-service-rate difference for DRAPE relative to each comparator. Filled points show means with paired 95% t intervals, and open circles show the ten completed seeds. Positive values in panels (a) and (c) favor DRAPE.

End-to-end result. Across ten paired seeds, DRAPE attains **4.0778** cost per request versus **4.1168–4.1540** and wins every seed. Its **1.04%–1.95%** reductions yield $p = 0.00195$ and Holm-adjusted $p = 0.00977$ for every comparison, with delivery savings exceeding added cache-update and recommendation costs. DRAPE records **92.60** ms latency and **26.71%** edge service, compared with OOA at **86.20** ms and **33.45%**, so the advantage is confined to the optimized cost. Removing cost guidance adds 0.0452 across five diagnostic seeds, whereas personality ablations show no resolved difference, supporting projection but not a separate personality effect. Appendices G–K report the mapping, stress, scaling, execution, and subgroup diagnostics.

6 Conclusion

In this paper, we investigate personality-aware joint recommendation and caching in D2D-assisted edge networks. We propose RITE to infer Big Five profiles from sparse rating histories and combine them with recommendation-list context to estimate user-specific responses. We further develop DRAPE to coordinate recommendation and caching through centralized training, decentralized execution, and cost-guided action projection. Trace-driven experiments show that DRAPE reduces total system cost against five scenario-adapted baselines, while ablations support the contribution of cost-guided projection. Future work will examine the behavioral contribution of inferred personality using data that jointly record recommendation responses and network interactions.

Ethics statement. Personality inference is sensitive. Deployment requires a legal basis, data minimization, access control, opt-out, provenance and license documentation, and subgroup audits (Han et al., 2025; Zhao et al., 2025b). Predicted traits are not identities.

Reproducibility statement. Code and Appendix L record preprocessing, splits, hyperparameters, seeds, software versions, fitted models, results, and figure source data. An anonymous archive will be provided at submission.

AI use statement. OpenAI Codex assisted restructuring, drafting, editing, and LaTeX preparation, but did not generate observations or select results. The authors take responsibility for all claims, citations, code, and outputs.

REFERENCES

Shihao Cai, Jizhi Zhang, Keqin Bao, Chongming Gao, Qifan Wang, Fuli Feng, and Xiangnan He. Agentic feedback loop modeling improves recommendation and user simulation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2235–2244, 2025. doi: 10.1145/3726302.3729893.

C. L. Philip Chen and Zhulin Liu. Broad learning system: An effective and efficient incremental learning system without the need for deep architecture. *IEEE Transactions on Neural Networks and Learning Systems*, 29(1):10–24, 2018. doi: 10.1109/TNNLS.2017.2716952.

Siran Chen, Boyu Chen, Chenyun Yu, Yi Ouyang, Lei Cheng, Chengxiang Zhuo, Zang Li, and Yali Wang. When top-ranked recommendations fail: Modeling multi-granular negative feedback for explainable and robust video recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(24):20262–20270, 2026a. doi: 10.1609/aaai.v40i24.39114.

Xiangchun Chen, Jiannong Cao, Rui Cao, Yuvraj Sahni, Mingjin Zhang, and Yusheng Ji. Decentralized task offloading in collaborative edge computing: A digital twin assisted multi-agent reinforcement learning approach. *IEEE Transactions on Mobile Computing*, 25(4):4776–4790, 2026b. doi: 10.1109/TMC.2025.3628502.

Xiangyi Chen, Yuanguo Bi, Juan Zhang, Xiaoming Yuan, Dusit Niyato, Liang Zhao, and Xingwei Wang. Task offloading and resource optimization based on dependency-aware graph and collaborative deep reinforcement learning in mobile edge computing. *IEEE Transactions on Mobile Computing*, 25(7):10998–11015, 2026c. doi: 10.1109/TMC.2026.3665729.

Zheyi Chen, Jiayu Zheng, Hongju Cheng, Geyong Min, Zhaolong Ning, Jie Li, and Yan Zhang. Joint service caching and resource allocation in DT-empowered cloud-edge networks via MARL with hierarchical knowledge transfer. *IEEE Transactions on Mobile Computing*, 25(10):17243–17261, 2026d. doi: 10.1109/TMC.2026.3691308.

Xingxia Dai, Shujuan Tian, Haolin Liu, Zhetao Li, Hongbo Jiang, and Qingyong Deng. Joint optimization of offloading and caching in full-duplex-enabled edge computing networks. *IEEE Transactions on Mobile Computing*, 24(8):6996–7011, 2025. doi: 10.1109/TMC.2025.3546263.

Qingyong Deng, Mengyao Li, Zhetao Li, Haolin Liu, and Yong Xie. End-edge collaborative optimization of microservice caching in D2D-assisted network. *IEEE Transactions on Mobile Computing*, 24(11):12162–12176, 2025. doi: 10.1109/TMC.2025.3582579.

Chongwu Dong, Weidong Li, Zhi Zhou, Xu Chen, Zhihong Tian, and Wushao Wen. Delay-sensitive task offloading with edge caching through martingale-based deep reinforcement learning. *IEEE Transactions on Mobile Computing*, 24(7):6225–6242, 2025. doi: 10.1109/TMC.2025.3540413.

Mingzhe Han, Dongsheng Li, Jiafeng Xia, Jiahao Liu, Hansu Gu, Peng Zhang, Ning Gu, and Tun Lu. FedCIA: Federated collaborative information aggregation for privacy-preserving recommendation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1687–1696, 2025. doi: 10.1145/3726302.3729977.

Joshua Holder, Natasha Jaques, and Mehran Mesbahi. Multi agent reinforcement learning for sequential satellite assignment problems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(25):26516–26524, 2025. doi: 10.1609/aaai.v39i25.34852.

Zheyuan Hu, Jianwei Niu, Tao Ren, Xuefeng Liu, and Mohsen Guizani. SITOff: Enabling size-insensitive task offloading in D2D-assisted mobile edge computing. *IEEE Transactions on Mobile Computing*, 24(3):1567–1584, 2025. doi: 10.1109/TMC.2024.3483951.

Zhaoxiang Huang, Zhiwen Yu, Liang Wang, Yingnan Zhao, Huan Zhou, and Bin Guo. Two time-scale DRL for service caching and task offloading in cross-domain marine networks. *IEEE Transactions on Mobile Computing*, 25(1):785–800, 2026. doi: 10.1109/TMC.2025.3594602.

Wei Hung, Shao-Hua Sun, and Ping-Chun Hsieh. Efficient action-constrained reinforcement learning via acceptance-rejection method and augmented MDPs. In *The Thirteenth International Conference on Learning Representations*, 2025.

Hongbo Jiang, Jianghao Cai, Zhu Xiao, Kehua Yang, Hongyang Chen, and Jiangchuan Liu. Vehicle-assisted service caching for task offloading in vehicular edge computing. *IEEE Transactions on Mobile Computing*, 24(7):6688–6700, 2025. doi: 10.1109/TMC.2025.3545444.

Hongbo Jiang, Jianghao Guo, Zhu Xiao, Jiali Yang, Kehua Yang, and Geyong Min. Cooperative content caching in vehicular edge computing networks: A two-stage deep reinforcement learning approach. *IEEE Transactions on Mobile Computing*, 25(7):10557–10576, 2026. doi: 10.1109/TMC.2026.3664597.

Jieyong Kim, Hyunseo Kim, Hyunjin Cho, SeongKu Kang, Buru Chang, Jinyoung Yeo, and Dongha Lee. Review-driven personalized preference reasoning with large language models for recommendation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1697–1706, 2025. doi: 10.1145/3726302.3730055.

Yuyuan Li, Junjie Fang, Fengyuan Yu, Xichun Sheng, Tianyu Du, Xuyang Teng, Shaowei Jiang, Linbo Jiang, Jianan Lin, and Chaochao Chen. FedAU2: Attribute unlearning for user-level federated recommender systems with adaptive and robust adversarial training. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(28):23310–23318, 2026. doi: 10.1609/aaai.v40i28.39500.

Yuzhu Liang, Haiyang Huang, Changfu Xu, Haodong Zou, Xinggang Fan, Yupeng Li, and Tian Wang. Cooperative edge caching for profit maximization based on multi-agent deep reinforcement learning. *IEEE Transactions on Mobile Computing*, pp. 1–14, 2026. doi: 10.1109/TMC.2026.3710512. Early Access.

Xiaoyu Liu, Jiaxin Yuan, Yuhang Zhou, Jingling Li, Furong Huang, and Wei Ai. CSRec: Rethinking sequential recommendation from a causal perspective. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1562–1571, 2025. doi: 10.1145/3726302.3729940.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Ruikun Luo, Zujia Zhang, Qiang He, Mengxi Xu, Feifei Chen, Xiaohai Dai, Song Wu, and Hai Jin. Cost-effective edge data caching with failure tolerance and popularity awareness. *IEEE Transactions on Mobile Computing*, 24(6):5357–5369, 2025. doi: 10.1109/TMC.2025.3531967.

Chenglong Ma, Ziqi Xu, Yongli Ren, Danula Hettiachchi, and Jeffrey Chan. PUB: An LLM-enhanced personality-driven user behaviour simulator for recommender system evaluation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2690–2694, 2025. doi: 10.1145/3726302.3730238.

Duong Thuy Anh Nguyen, Jiaming Cheng, Ni Trieu, Duong Tung Nguyen, and Angelia Nedić. Bi-CrowdCache: A decentralized game-theoretic model for edge content sharing over time-varying communication networks. *IEEE Transactions on Mobile Computing*, 25(2):2895–2907, 2026. doi: 10.1109/TMC.2025.3611963.

Shuting Qiu, Fang Dong, Siyu Tan, Dian Shen, Ruiting Zhou, and Qilin Fan. CoCaR: Enabling efficient dynamic DNN-based model caching and request routing in MEC. In *IEEE INFOCOM 2025—IEEE Conference on Computer Communications*, pp. 1–10, 2025. doi: 10.1109/INFOCOM55648.2025.11044457.

Yimin Shi, Yang Fei, Shiqi Zhang, Haixun Wang, and Xiaokui Xiao. You are what you bought: Generating customer personas for e-commerce applications. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1810–1819, 2025. doi: 10.1145/3726302.3730118.

Yuxuan Song, Qiudan Li, Yilin Wu, David Jingjun Xu, and Daniel Dajun Zeng. Knowledge-enhanced hierarchical heterogeneous graph for personality identification with limited training data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(2):1529–1537, 2025. doi: 10.1609/aaai.v39i2.32144.

Haiyang Sun, Honglong Chen, Xinglong Fan, Zhichen Ni, Liantao Wu, Peng Sun, and Weifeng Liu. MATE: A D2D-enhanced multi-bitrate video caching strategy for cloud-edge-device collaborative networks. *IEEE Transactions on Mobile Computing*, 25(5):6966–6980, 2026. doi: 10.1109/TMC.2025.3641174.

Lu Wang, Di Zhang, Fangkai Yang, Pu Zhao, Jianfeng Liu, Yuefeng Zhan, Hao Sun, Qingwei Lin, Weiwei Deng, Dongmei Zhang, Feng Sun, and Qi Zhang. LettinGo: Explore user profile generation for recommendation system. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 2985–2995, 2025. doi: 10.1145/3711896.3737024.

Tong Wu, Dongjin Yu, Chengfei Liu, Dongjing Wang, and Binbin Huang. Recommendation-enabled edge caching and D2D offloading via incentive-driven deep reinforcement learning. *IEEE Transactions on Services Computing*, 17(4):1724–1738, 2024. doi: 10.1109/TSC.2024.3351219.

Dianlei Xu, Xiang Su, Gopika Premsankar, Huandong Wang, Sasu Tarkoma, and Pan Hui. Dynamic hierarchical reinforcement learning framework for energy-efficient 5G base stations in urban environments. *IEEE Transactions on Mobile Computing*, 24(9):8582–8599, 2025. doi: 10.1109/TMC.2025.3557280.

Guang Yang, Tianpei Yang, Jingwen Qiao, Yanqing Wu, Jing Huo, Xingguo Chen, and Yang Gao. Multi-agent reinforcement learning with communication-constrained priors. In *Advances in Neural Information Processing Systems*, volume 38, pp. 170968–170988, 2025. doi: 10.52202/085713-5150.

Yuqin Yang, Haowu Zhou, Haoran Tu, Zhiwen Hui, Shiqi Yan, HaoYang Li, Dong She, Xianrong Yao, Yang Gao, and Zhanpeng Jin. Persona-E^2: A human-grounded dataset for personality-shaped emotional responses to textual events. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29290–29315, 2026. doi: 10.18653/v1/2026.acl-long.1350.

Chunxu Zhang, Guodong Long, Hongkuan Guo, Zhaojie Liu, Guorui Zhou, Zijian Zhang, Yang Liu, and Bo Yang. Multifaceted user modeling in recommendation: A federated foundation models approach. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(12):13197–13205, 2025a. doi: 10.1609/aaai.v39i12.33440.

Zijian Zhang, Shuchang Liu, Ziru Liu, Rui Zhong, Qingpeng Cai, Xiangyu Zhao, Chunxu Zhang, Qidong Liu, and Peng Jiang. LLM-powered user simulator for recommender system. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(12):13339–13347, 2025b. doi: 10.1609/aaai.v39i12.33456.

Liang Zhao, Lu Sun, Ammar Hawbani, Zhi Liu, Xiongyan Tang, and Lexi Xu. Dual dependency-aware collaborative service caching and task offloading in vehicular edge computing. *IEEE Transactions on Mobile Computing*, 24(10):10963–10977, 2025a. doi: 10.1109/TMC.2025.3573379.

Xuhao Zhao, Zhongrui Zhang, Yanmin Zhu, Zhaobo Wang, Wenze Ma, Jiadi Yu, and Feilong Tang. Social relation-level privacy risks and preservation in social recommender systems. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1728–1737, 2025b. doi: 10.1145/3726302.3730086.

Ao Zhou, Jiayi Guan, Li Shen, Fan Lu, Sanqing Qu, Junqiao Zhao, Ziqiao Wang, Ya Wu, and Guang Chen. CHPO: Constrained hybrid-action policy optimization for reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 73157–73185, 2025a. doi: 10.52202/085713-2201.

Huan Zhou, Hao Wang, Zhiwen Yu, Bin Guo, Mingjun Xiao, and Jie Wu. Federated distributed deep reinforcement learning for recommendation-enabled edge caching. *IEEE Transactions on Services Computing*, 17(6):3640–3656, 2024. doi: 10.1109/TSC.2024.3433579.

Ruiting Zhou, Yifeng Huang, Yufeng Wang, Lei Jiao, Haisheng Tan, Renli Zhang, and Libing Wu. User preference oriented service caching and task offloading for UAV-assisted MEC networks. *IEEE Transactions on Services Computing*, 18(2):1097–1109, 2025b. doi: 10.1109/TSC.2025.3536319.

APPENDIX GUIDE

Appendix A proves the NP-hardness of the joint optimization, and Appendix B proves the feasibility of the DRAPE projection. Appendices C–K report dataset construction, experimental protocols, complete results, predictive diagnostics, robustness tests, ablations, scaling measurements, and subgroup analyses. Appendix L records the materials required to reproduce the experiments.

A NP-HARDNESS OF THE JOINT OPTIMIZATION

We reduce the 0–1 knapsack problem to a restricted instance of Eq. (7). Consider items $f \in \mathcal{F}$ with positive integer weights w_f, nonnegative values v_f, capacity W, and at least one positive value. Construct an instance with one slot, one user u, and one SBS m. Set $T = 1$, $r_u^{\mathrm{res},1} = 0$, $\lambda_r = 0$, and $D_{u,f}^1 = 0$ for every content, so recommendation is objective-neutral, the cache-update term is absent, and D2D delivery is disabled. Choose the instance parameters as follows:

$$L = \max_{f \in \mathcal{F}} v_f, \qquad s_f = w_f, \qquad C_m^1 = W, \qquad q_{u,f} = \frac{v_f}{L w_f}, \qquad \alpha_e = L. \tag{27}$$

Because $v_f \le L$ and $w_f \ge 1$, every $q_{u,f}$ lies in $[0, 1]$. Caching content f avoids backhaul cost $q_{u,f} s_f \alpha_e = v_f$. All remaining terms are independent of the cache decision, so the objective has the following cache-dependent form:

$$C_{\mathrm{const}} - \sum_{f \in \mathcal{F}} v_f x_{m,f}^1. \tag{28}$$

Minimizing this expression subject to the cache constraint is therefore equivalent to the 0–1 knapsack problem:

$$\begin{aligned} \max_{\{x_{m,f}^1\}} \quad & \sum_{f \in \mathcal{F}} v_f x_{m,f}^1 \\ \text{s.t.} \quad & \sum_{f \in \mathcal{F}} w_f x_{m,f}^1 \le W, \qquad x_{m,f}^1 \in \{0, 1\}. \end{aligned} \tag{29}$$

The construction is polynomial in the knapsack input size. Hence, the joint optimization in Eq. (7) is NP-hard. Unless $\mathrm{P} = \mathrm{NP}$, no polynomial-time algorithm can return a globally optimal decision for every instance.

B PROOF OF PROJECTION FEASIBILITY

Proposition 1 (Projection feasibility)**.** *Algorithm 2 returns decisions satisfying constraints* (C1)–(C3) *for every actor output when* $|\mathcal{C}^t| \ge K$.

Proof. For each SBS, every candidate is examined and admitted only when its size does not exceed the residual capacity, so the final cached size is at most C_m^t. Because $|\mathcal{C}^t| \ge K$, the top-K step selects exactly K distinct candidates for every user. Both outputs are binary by construction. The score values affect the ordering but not these feasibility checks. □

Table 2: Audited source-data counts after deterministic preprocessing.

Source	Raw/retained rows	Corrected/removed	Distinct entities
Personality profiles	1,834	14 duplicate rows removed	1,820 users
Personality ratings	1,028,751	0 removed	35,196 movies
INFOCOM train	50,000	31,265 zero durations corrected	78 users
INFOCOM test	78,901	42,789 zero durations corrected	78 users

C Synthetic Dataset Construction and Leakage Control

C.1 Source-data audit

The INFOCOM columns are sender, receiver, contact start, contact end, and an unused sequence identifier. A contact with identical start and end times is assigned a duration of 120 seconds, as specified by the data protocol. No negative-duration or self-contact rows were retained or removed because none were observed after parsing. The training trace spans timestamps 6,207–72,272, and the test trace spans 66,210–340,927. Their raw time ranges overlap, but the supplied train and test files remain distinct and are never merged for model selection.

C.2 User mapping, contents, and behavioral features

All INFOCOM identifiers are sorted, and 78 distinct Personality users are sampled uniformly without replacement using seed 2027. The sampled users are paired one-to-one with the INFOCOM identifiers. This fixed mapping creates a controlled synthetic benchmark; it does not imply that the two datasets describe the same real individuals. The 78 mapped users are excluded from fitting RITE. The remaining 1,742 users form the development pool, with 348 users used to select the incremental BLS learner within RITE before refitting on the full development pool. The pool is partitioned into 80% fitting users and 20% validation users. The search considers feature-node counts in $\{24, 48, 96\}$, total enhancement-node counts in $\{32, 64, 128\}$, and ridge coefficients in $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$. Enhancement nodes are appended in blocks of 32, so the three total widths correspond to $L \in \{1, 2, 4\}$. The setting with the lowest validation RMSE is selected, with MAE breaking ties. The selected incremental BLS learner within RITE uses 24 feature nodes, 128 enhancement nodes, ridge coefficient 1.0, and random seed 2044.

The content universe contains the 100 most frequently observed movies among mapped users. Missing user–movie ratings are completed with regularized user and item biases. The behavioral descriptor uses rating mean, rating variance, popularity-stratum entropy, popularity-stratum breadth, log rating activity, and rating extremity. For the reported experiments, we set $K_p = 10$. Entropy and breadth are computed over the resulting popularity strata estimated from development users.

C.3 Response construction and information boundaries

The policy-visible response model is fit out of fold. Its inputs are list context and RITE-inferred traits, and its output is a normalized estimate of the reported `enjoy_watching` field. The hidden simulator uses the mapped user's normalized reported value only when sampling requests. The same hidden response values and inverse-CDF request uniforms are shared by every method in a paired seed.

The only exception is the explicitly labelled oracle diagnostic, which exposes true traits and the true response to the policy to test an optimistic upper bound. It is excluded from the formal comparator table. Array-level audits confirm that adding the empirical response layer did not alter the original contact or preference arrays.

Table 3: Information boundary used to prevent target leakage and asymmetric evaluation.

Quantity	Fitting/tuning	Policy or projection	Hidden simulator
Development-user ratings and traits	Yes	No	No
Mapped-user rating history	Features only	Yes	Yes
RITE-inferred traits	Yes	Yes, for every learned method	No
Cross-fitted response prediction	Yes	Yes	No
Reported mapped-user `enjoy_watching`	No	No	Yes
INFOCOM training contacts	Yes	State construction	Yes
INFOCOM test contacts	No	Evaluation state only	Evaluation only
Paired request uniforms	No	No	Yes, shared across methods

Table 4: Configuration of the formal ten-seed comparison.

Environment	Value	Learning	Value
Users / contents	78 / 100	Training episodes	120
SBS agents	5	Evaluation episodes	30
Candidates / top-K	10 / 3	Episode horizon	20 slots
Slot duration	120 s	Actor / critic LR	2×10^{-4} / 5×10^{-4}
Device cache	5 items	Discount γ	0.95
SBS cache fraction	0.15	Target rate τ	0.01
D2D / SBS / cloud delay	8 / 25 / 120 ms	Batch / replay size	128 / 50,000
D2D / cellular / cloud cost	0.2 / 1.0 / 5.0	Warm-up steps	300
λ_r / λ_c	0.03 / 0.05	Updates per step	1
κ_x / κ_y	0.5 / 0.0	Hidden layers	256 / 128

Table 5: Comparator adaptations. Each method receives the same data, hidden request world, paired random numbers, action constraints, and training budget.

Method	Adaptation used in the common edge environment
EC-MADRL	CTDE MADDPG with a network-wide cooperative cost and population-mean response; continuous actions use neutral feasibility repair.
OOA	Greedy placement from preference and contact centrality, with cache revision when realized hit rate falls 0.05 below expectation.
DRA-SAC	Centralized entropy-regularized SAC using cache state and dynamic request history; cloud fallback represents delay risk in the common objective.
COHCTO	PPO with a gated graph-recurrent service-criticality proxy and active–passive hierarchical cache projection.
uEDC-L	Affine robust scoring in nominal demand, demand deviation, and contact-outage risk.

D Experimental Protocol and Implementation

D.1 Environment and optimization settings

Five SBS groups are obtained by spectral clustering of aggregate training-trace contact duration. The candidate pool combines high-preference, uncertain, popular, and exploratory contents so that every policy acts on the same reduced action universe. Costs in Table 4 are per content-size unit. Cloud retrieval consists of cellular cost 1.0 plus backhaul cost 4.0, yielding total cost 5.0. Exploration noise is applied only to the pre-projection continuous scores and decays linearly from 0.30 to 0.03 during training.

Table 6: Complete test summary over ten training/environment seeds (mean ± SD). Lower is better for cost and latency; higher is better for service rates.

Method	Cost/request ↓	Latency (ms) ↓	Cache hit (%) ↑	D2D offload (%) ↑	Edge service (%) ↑
DRAPE	**4.0778 ± 0.5192**	92.60 ± 11.62	14.80 ± 0.97	11.90 ± 11.12	26.71 ± 10.25
EC-MADRL	4.1381 ± 0.4982	88.79 ± 10.43	18.79 ± 2.20	11.92 ± 11.11	30.71 ± 8.99
OOA	4.1168 ± 0.4925	**86.20 ± 9.83**	**21.52 ± 2.75**	**11.93 ± 11.10**	**33.45 ± 8.36**
DRA-SAC	4.1343 ± 0.4940	88.75 ± 10.29	18.83 ± 2.28	11.93 ± 11.09	30.76 ± 8.84
COHCTO	4.1280 ± 0.4915	86.56 ± 9.99	21.15 ± 2.58	11.92 ± 11.08	33.07 ± 8.54
uEDC-L	4.1540 ± 0.4857	89.21 ± 9.66	18.35 ± 2.94	11.93 ± 11.11	30.28 ± 8.18

Table 7: Paired cost improvement of DRAPE over each comparator. Positive values favor DRAPE. Confidence intervals are two-sided 95% t intervals across the ten paired seeds.

Comparator	Mean improvement	95% CI	Wins/10	Wilcoxon p / Holm p
EC-MADRL	0.0602	[0.0447, 0.0757]	10	0.00195 / 0.00977
OOA	0.0390	[0.0189, 0.0590]	10	0.00195 / 0.00977
DRA-SAC	0.0564	[0.0374, 0.0754]	10	0.00195 / 0.00977
COHCTO	0.0502	[0.0285, 0.0718]	10	0.00195 / 0.00977
uEDC-L	0.0762	[0.0510, 0.1013]	10	0.00195 / 0.00977

D.2 Scenario-adapted comparators

The labels in Table 5 denote transparent scenario adaptations rather than exact author-code reproductions. Learned comparators retain their own policy families and feasibility decoders. They are not passed through DRAPE's cost-guided projection. The fair-comparison audit also checks that the method label cannot change the hidden response model, contact process, candidate set, or request uniforms.

D.3 Statistical protocol

The formal comparison uses seeds 2027–2036. A complete training/environment seed is the independent unit; episodes, slots, users, and contact events are repeated observations within that unit. We report two-sided 95% t intervals across seeds and paired two-sided Wilcoxon signed-rank tests. Holm correction is applied across the five method comparisons within each metric. Seeds 2032–2036 were fixed before the extension run and analyzed only after the batch completed. We additionally report paired nonparametric bootstrap intervals, exact sign tests, standardized paired effects, leave-one-seed-out means, and an omnibus Friedman test; these reuse the same ten independent units and are robustness checks rather than additional replications. The five-seed ablation and stress experiments are interpreted through direction and effect magnitude because their minimum attainable two-sided signed-rank value is 0.0625 when every difference has the same sign. In the mapping study, three training seeds are nested within each of five independently rebuilt mappings; inference is performed on the five mapping-level means, never on the 15 nested runs. The three-seed scaling and subgroup studies and all timing blocks are descriptive.

E Full End-to-End Results

Table 6 makes the multi-objective boundary visible. DRAPE has the lowest stated cost but not the lowest latency or the highest edge-service rate. D2D offload is nearly unchanged across methods because the same contact world and helper availability dominate that route. The differentiating behavior is primarily the allocation between cache hits and cloud fallback, together with cache-update and recommendation penalties.

The paired design is substantially more informative than overlapping marginal intervals in Table 6. Contact realizations make the between-seed SD roughly 0.49–0.52 cost units, but the common hidden world removes much of that variation from within-seed method differences. This produces positive paired intervals and ten of ten lower-cost outcomes for every comparator.

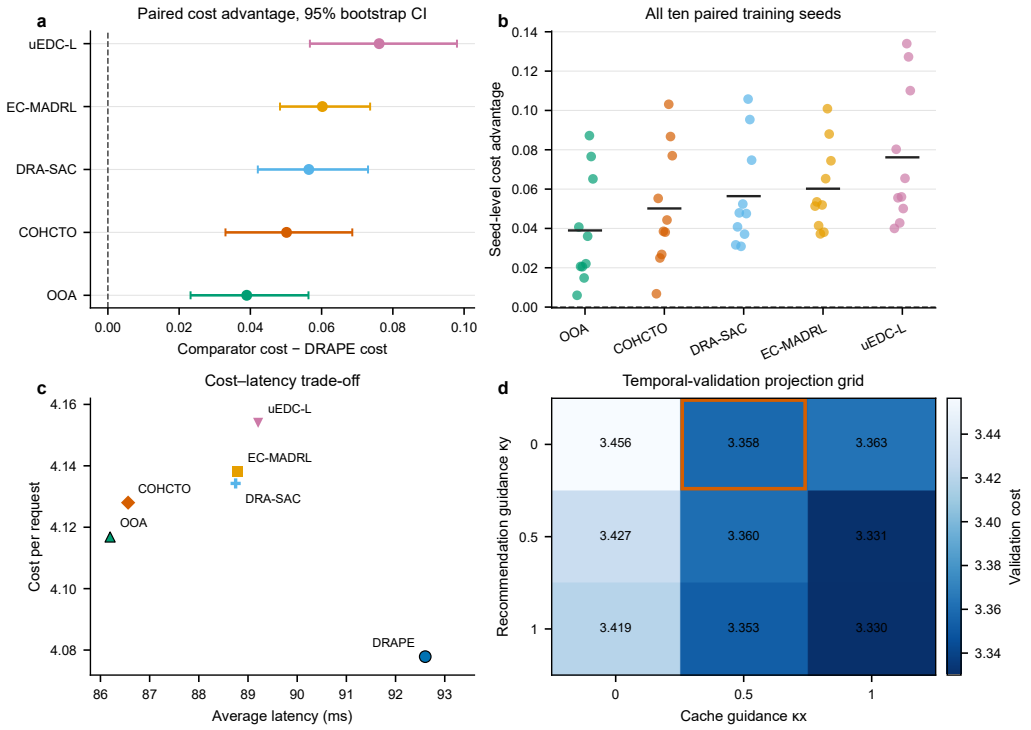

Figure 3: Robustness of the formal paired cost result. Panel (a) shows paired bootstrap mean cost advantages with 95% intervals. Panel (b) shows the ten seed-level paired advantages with horizontal mean bars. Panel (c) shows mean cost–latency operating points. Panel (d) shows temporal-validation cost over cache and recommendation guidance weights κ_x and κ_y, with the selected setting outlined. Positive advantages favor DRAPE.

Table 8: Decomposition of paired cost improvement per request. Positive entries favor DRAPE; negative entries are extra costs paid by DRAPE. Components sum to the total paired improvement up to rounding.

Comparator	Delivery	Cache update	Recommendation	Total
EC-MADRL	0.0698	-0.0091	-0.0005	0.0602
OOA	0.0621	-0.0090	-0.0142	0.0390
DRA-SAC	0.0658	-0.0090	-0.0004	0.0564
COHCTO	0.0640	-0.0083	-0.0056	0.0502
uEDC-L	0.0821	0.0109	-0.0169	0.0762

The paired bootstrap intervals remain above zero: $[0.0483, 0.0736]$ for EC-MADRL, $[0.0232, 0.0563]$ for OOA, $[0.0421, 0.0730]$ for DRA-SAC, $[0.0330, 0.0686]$ for COHCTO, and $[0.0567, 0.0980]$ for uEDC-L. Exact sign and sign-flip tests give $p = 0.001953$ for every comparator and Holm-adjusted $p = 0.009766$; standardized paired effects range from $d_z = 1.39$ to 2.78. Every leave-one-seed-out mean also stays positive. Across all six methods, the Friedman test gives $\chi^2(5) = 43.66$, $p = 2.72 \times 10^{-8}$, and Kendall's $W = 0.873$; DRAPE ranks first in all ten seeds, followed by OOA. Cost and latency do not collapse to one ordering: only DRAPE and OOA are nondominated at the mean operating points in figure 3c.

Table 8 identifies the source of the objective advantage. DRAPE saves 0.062–0.082 delivery-cost units per request. Against four methods, it spends about 0.008–0.009 more on cache replacement; against OOA, COHCTO, and uEDC-L, it also spends more on recommendation. These penalties are smaller than the delivery saving. The result therefore reflects deliberate expenditure to reduce expensive delivery rather than uniformly better behavior on every component.

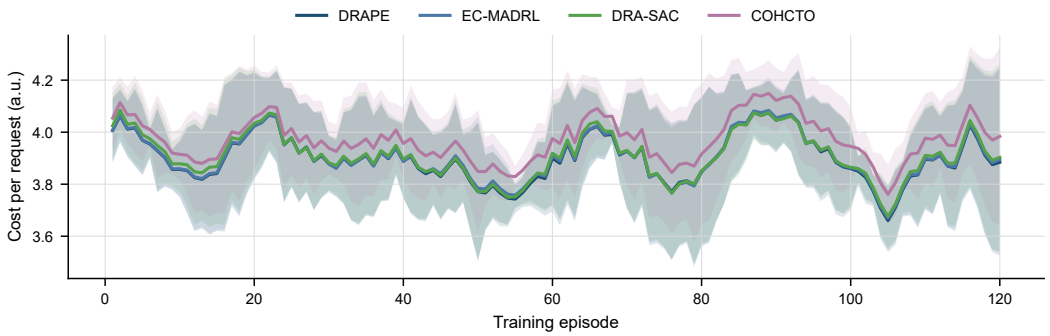

Figure 4: Training cost per request for learned methods. Curves show the mean 11-episode centered moving average across ten complete seeds; shading is ± 1.96 standard errors and is descriptive rather than an inferential interval. Formal summaries use unsmoothed test episodes.

Table 9: Held-out Big Five prediction on the 78 mapped users. Scores are normalized to $[0, 1]$. The training-mean predictor is fit without using mapped-user labels.

Trait	RITE MAE	Mean MAE	RITE RMSE	Mean RMSE	RITE Pearson r
Openness	0.1334	0.1342	0.1610	0.1610	0.143
Conscientiousness	0.2131	0.2190	0.2547	0.2575	0.153
Extraversion	0.2023	0.2104	0.2479	0.2546	0.234
Agreeableness	0.1743	0.1745	0.2133	0.2107	0.088
Emotional stability	0.2136	0.2060	0.2477	0.2401	-0.080
Macro	0.1873	0.1888	0.2276	0.2276	0.108

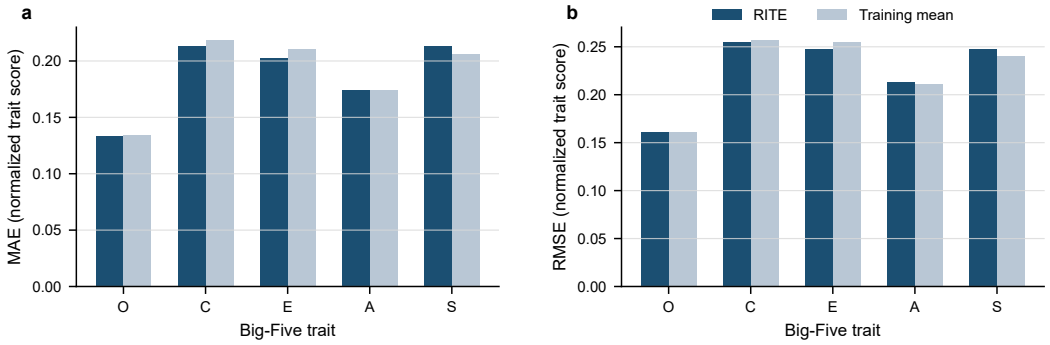

Figure 5: Held-out Big Five prediction error for 78 mapped users. Traits are openness (O), conscientiousness (C), extraversion (E), agreeableness (A), and emotional stability (S).

The same trade-off appears in operational metrics. Relative to EC-MADRL, OOA, DRA-SAC, COHCTO, and uEDC-L, DRAPE incurs mean latency penalties of 3.81, 6.41, 3.86, 6.04, and 3.40 ms, respectively. Its edge-service rate is lower by 4.01, 6.74, 4.05, 6.36, and 3.57 percentage points. All paired latency and edge-service comparisons have Holm-adjusted $p = 0.00977$. The optimization objective therefore should not be read as a proxy for latency dominance.

The learning curves are used only to inspect optimization behavior. They do not create additional independent samples, and visual smoothness is not treated as evidence of statistical significance. OOA and uEDC-L have no learned training curve and are therefore absent from figure 4.

Table 10: Five independently reconstructed mappings. Δ response MAE is personality-conditioned minus context-only MAE in units of 10^{-3}, so negative values favor the personality-conditioned model. Costs are mapping-level means over three nested training seeds.

Mapping seed	RITE MAE	Δ response MAE	DRAPE	EC-MADRL	OOA	uEDC-L	DRAPE rank
2027	0.1873	$+1.407$	**4.0033**	4.0606	4.0338	4.0911	1
2028	0.1779	-0.228	**4.2138**	4.2686	4.2674	4.2868	1
2029	0.1662	$+1.414$	**3.9440**	4.0211	4.0590	4.0680	1
2030	0.1809	$+0.130$	**4.1741**	4.2401	4.2143	4.2446	1
2031	0.1758	-1.157	**4.2350**	4.3571	4.3179	4.3796	1

F RITE AND RESPONSE DIAGNOSTICS

RITE's improvement over a constant training-mean predictor is small. Its macro MAE decreases by 0.0015, while macro RMSE is effectively unchanged. Correlations are weak and emotional stability has a slightly negative correlation. A user-level bootstrap gives a 95% interval of [0.1741, 0.2018] for macro MAE and [0.2011, 0.2320] for user-level RMSE. These checks justify treating the representation as noisy side information rather than an accurate psychological measurement.

Across all 1,820 Personality users, the list-context-only ridge model has raw-scale MAE 0.8808, RMSE 1.0526, and Pearson correlation 0.0865. Adding RITE-inferred traits changes these values to 0.8789, 1.0516, and 0.0970, respectively, for a paired MAE improvement of 0.00193. On the 78 mapped users, however, normalized MAE is 0.1373 with personality and 0.1359 without it. The population-level change therefore does not transfer as a measurable advantage to the mapped evaluation group. This mismatch motivates the stronger ablation boundary in Appendix H.

G ALTERNATIVE-MAPPING ROBUSTNESS

The formal ten-seed comparison fixes mapping seed 2027 so that every method encounters exactly the same users and contacts. To measure sensitivity to that arbitrary pairing, we independently repeat user sampling, train/validation exclusion, RITE selection and refitting, cross-fitted response prediction, content selection, missing-rating completion, and one-to-one INFOCOM assignment for mapping seeds 2027–2031. DRAPE, EC-MADRL, OOA, and uEDC-L are then evaluated with three nested training seeds per mapping on a diagnostic configuration with 20 users, 30 contents, three agents, six candidates, top-$K = 2$, 50 training episodes, 20 evaluation episodes, and a horizon of ten slots. This design yields 60 complete runs. The mapping-level mean, not each nested run, is the independent unit.

DRAPE is lower in mean cost than every comparator in every mapping. Its mapping-level advantages are 0.0755 over EC-MADRL (bootstrap 95% interval $[0.0581, 0.0996]$), 0.0645 over OOA ($[0.0390, 0.0936]$), and 0.1000 over uEDC-L ($[0.0750, 0.1262]$). Five aligned differences yield two-sided Wilcoxon $p = 0.0625$ in each comparison, so the direction is consistent but not conventionally resolved. The response model is less stable: adding inferred personality improves mapped-user MAE in only two mappings and worsens it in three. Remapping therefore strengthens the cost observation associated with DRAPE's projection while weakening any interpretation that requires a reliable personality-response increment.

H DRAPE PROJECTION SELECTION AND COMPONENT ABLATIONS

H.1 TEMPORAL VALIDATION OF GUIDANCE WEIGHTS

Projection weights are selected without accessing the INFOCOM test trace. The training trace is split chronologically at 58,946 seconds, producing 39,665 early training contacts and 10,006 late validation contacts; 329 contacts crossing the boundary are excluded. A 3×3 grid is evaluated with seeds 2027–2029 on the diagnostic configuration.

The minimum mean occurs at $(1.0, 1.0)$, but its standard error is 0.1167. Multiple weaker settings lie inside that uncertainty band. The predeclared rule therefore selects $(0.5, 0.0)$ because it has

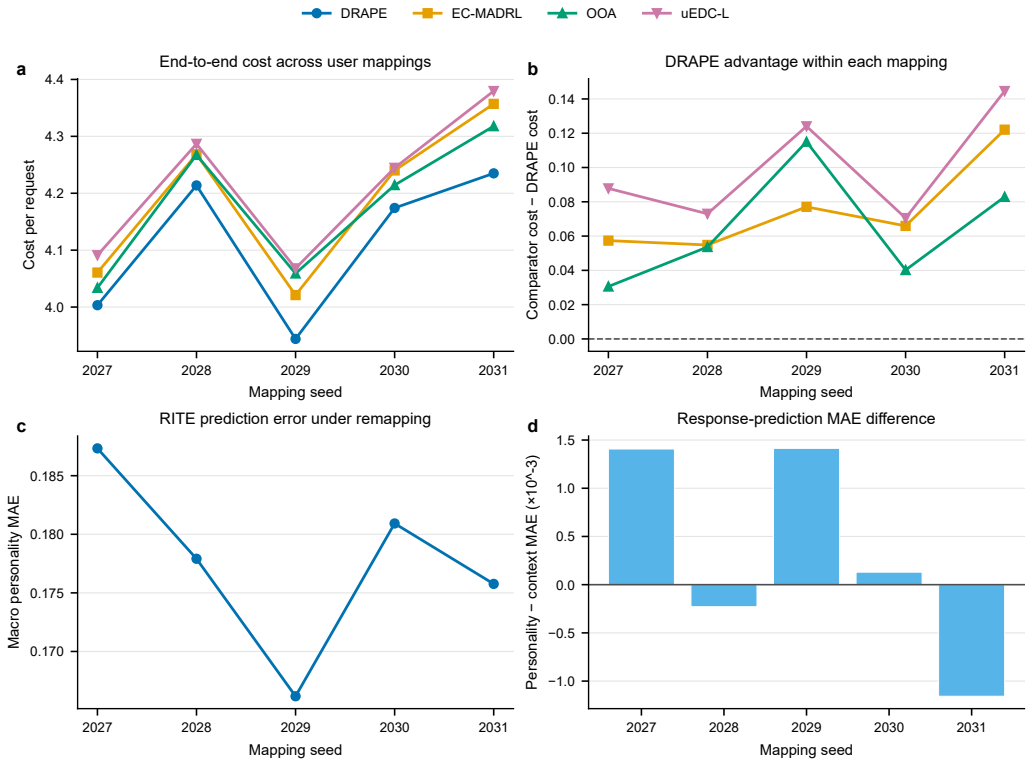

Figure 6: Sensitivity to cross-dataset user mapping. Panel (a) shows mapping-level end-to-end costs averaged over three nested training seeds. Panel (b) shows comparator cost minus DRAPE cost for each mapping. Panel (c) reports RITE prediction error under remapping. Panel (d) reports the personality-conditioned minus context-only response MAE.

Table 11: Validation cost for projection weights (mean ± SD across three seeds). The minimum is not automatically selected: the one-standard-error rule chooses the least guided setting within one SE of the minimum.

κ_x	κ_y	Cost/request	Latency (ms)	Decision
1.0	1.0	3.3300 ± 0.2021	77.68	Minimum mean
1.0	0.5	3.3308 ± 0.1926	78.01	Within one SE
0.5	1.0	3.3526 ± 0.1591	78.51	Within one SE
0.5	0.0	3.3577 ± 0.1774	78.46	**Selected**
0.5	0.5	3.3601 ± 0.1466	78.72	Within one SE
1.0	0.0	3.3626 ± 0.1423	78.23	Within one SE
0.0	1.0	3.4189 ± 0.1522	80.17	Within one SE
0.0	0.5	3.4275 ± 0.1484	80.11	Within one SE
0.0	0.0	3.4564 ± 0.1364	81.23	Outside one SE

the smallest total guidance strength among eligible settings, with latency and cost used only as tie-breakers. A candidate two-pass joint projection was also screened on validation data. Its mean cost was 3.3762 compared with 3.3577 for the selected projection, so it was rejected before formal testing.

H.2 Paired mechanism diagnostics

Removing cost guidance from the projection in DRAPE-NoGuide increases mean cost by 0.0452 per request and worsens all five paired seeds. With $n = 5$, an all-positive signed-rank result still gives $p = 0.0625$, so this diagnostic is consistent and practically sized but not conventionally significant.

Table 12: Five-seed diagnostic ablation. "Difference" is variant cost minus DRAPE-Full cost, so positive values favor DRAPE-Full. DRAPE-Full uses RITE-inferred traits and predicted response. DRAPE-Mean uses mean traits and mean response. DRAPE-Shuffle jointly permutes traits and response across users. DRAPE-NoGuide sets $\kappa_x = \kappa_y = 0$ while retaining feasibility projection. DRAPE-Oracle uses true traits and true response. The Wilcoxon tests are uncorrected diagnostic tests.

Variant	Cost/request	Difference	Wins/5	Wilcoxon p
DRAPE-Full	4.1273 ± 0.5111	–	–	–
DRAPE-Mean	4.1180 ± 0.5131	-0.0093	1	0.4375
DRAPE-Shuffle	4.1328 ± 0.4912	0.0055	3	0.8125
DRAPE-NoGuide	4.1725 ± 0.5084	0.0452	5	0.0625
DRAPE-Oracle	4.1283 ± 0.5061	0.0010	2	0.8125

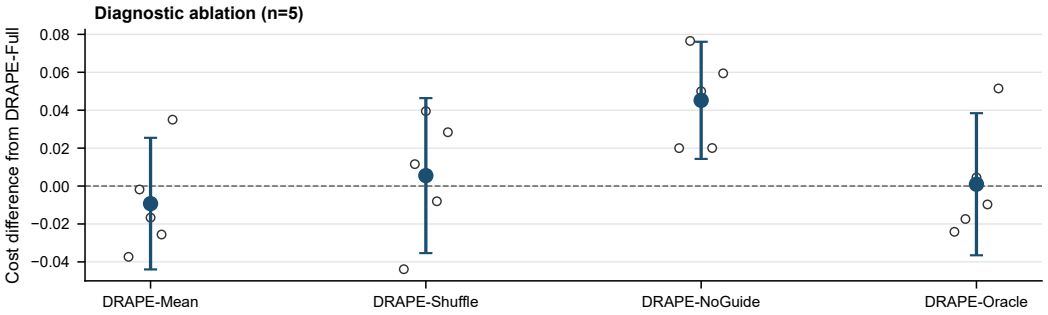

Figure 7: Paired diagnostic ablation on five seeds. DRAPE-Mean, DRAPE-Shuffle, DRAPE-NoGuide, and DRAPE-Oracle respectively use mean traits and response, jointly permuted traits and response, zero cost-guidance weights with feasibility projection retained, and true traits and response. Filled points are mean cost differences relative to DRAPE-Full, open circles are individual-seed differences, and error bars are paired 95% t intervals. Positive values favor DRAPE-Full. DRAPE-NoGuide is higher in every seed, while the other variants do not yield a resolved difference.

The personality perturbations provide a different message: DRAPE-Mean, DRAPE-Shuffle, and DRAPE-Oracle never produce a stable improvement over DRAPE-Full. In particular, DRAPE-Oracle rules out the explanation that prediction noise alone hides a large downstream gain under this simulator and budget. The experiment therefore supports DRAPE's cost-guided projection as the active component while leaving the person-specific modeling contribution unverified.

I Operating-Condition Stress Tests

The formal experiment retrains and evaluates every method under one specified environment. We additionally ask whether the learned ordering survives parameter shift without retraining. DRAPE and EC-MADRL are trained once under the baseline configuration below; OOA and uEDC-L are instantiated under the same baseline information. The four methods are then evaluated on identical paired stochastic inputs for five seeds under 32 conditions: the baseline plus 31 single-factor perturbations. This produces $32 \times 4 \times 5 = 640$ complete evaluations.

Figures 8 and 9 show the complete sweep. At the stress-suite baseline, mean costs are 4.6285 for DRAPE, 4.6761 for OOA, 4.6965 for EC-MADRL, and 4.7157 for uEDC-L. As expected, more cache reduces cost, larger backhaul cost raises it, and removing D2D contacts is harmful. Response scaling, behavior noise, and list length have smaller effects under this simulator, which is consistent with the weak behavioral signal observed earlier. Across 32 conditions and three comparators, DRAPE's mean cost advantage is positive in all 96 pairs and has the same favorable sign in every seed for 74. The narrowest margin is 0.0088 against OOA at $\lambda_r = 0.12$, where DRAPE wins four of five seeds and the bootstrap interval crosses zero. No paired test survives within-sweep Holm correction. The stress study therefore locates an operating range in which the ordering is persistent, but it does not turn five diagnostic seeds into confirmatory evidence.

Table 13: Zero-shot operating-condition audit. One factor changes at a time while learned weights remain fixed. Contact fraction multiplies retained D2D contact opportunities; response scale multiplies hidden recommendation response; behavior noise is Gaussian noise added to policy-visible behavioral estimates.

Factor	Training baseline	Test alternatives
SBS cache fraction	0.15	0.05, 0.10, 0.20, 0.30
Backhaul cost	4	2, 6, 8
Retained D2D contacts	1.00	0, 0.25, 0.50, 0.75
Hidden response scale	1.00	0, 0.50, 1.50
Behavior-model noise SD	0	0.05, 0.10, 0.20
Recommendation list K	3	1, 2, 5
Recommendation penalty λ_r	0.03	0, 0.06, 0.12
Cache-update penalty λ_c	0.05	0, 0.10, 0.20
Slot duration	120 s	60, 240, 480 s
Preference log-odds temperature	1.00	0.50, 2.00

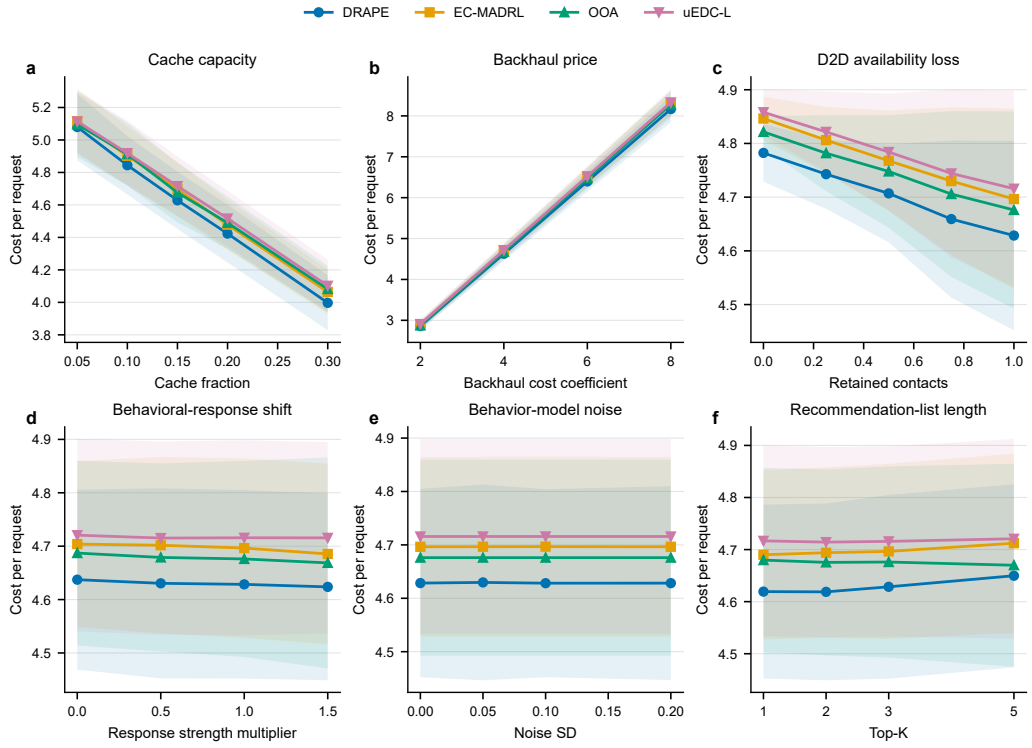

Figure 8: Zero-shot stress results for core network and behavioral shifts. Lines are five-seed means and bands are 95% t intervals. The dashed vertical position in each panel identifies the training baseline when it belongs to the displayed range.

J Retrained Scaling and Measured Execution

To separate zero-shot shift from growth in state and action dimensions, we retrain each learned method at three sizes. The small, medium, and full settings use 20/30/3, 40/60/4, and 78/100/5 users/contents/agents, respectively; candidate counts are 6, 8, and 10, while K is 2, 3, and 3. Each setting uses three complete training seeds, 45 training episodes, 15 evaluation episodes, and a ten-slot horizon. OOA and uEDC-L are evaluated with the same paired worlds. Training time and decision latency are hardware measurements, not algorithm-independent complexity guarantees.

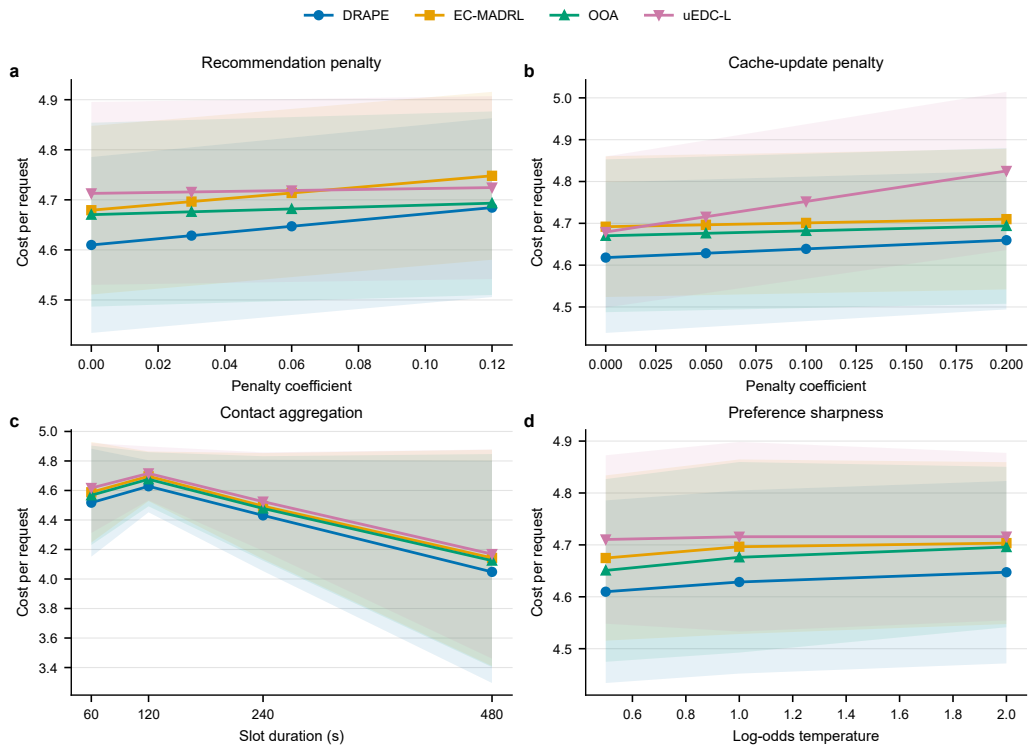

Figure 9: Zero-shot stress results for recommendation penalty, cache-update penalty, slot duration, and preference log-odds temperature. All methods use the same shifted environment and paired random inputs. Learned weights are not updated.

Table 14: Retrained scale study and measured execution. Cost is averaged over three training seeds. DRAPE training time covers 45 episodes. Decision entries are medians over ten timing blocks of 50 calls after ten warm-up calls. Parameter memory assumes float32; GPU peak allocation was measured on an RTX 4060 Ti.

Scale	DRAPE cost	EC-MADRL cost	OOA cost	uEDC-L cost	DRAPE train (s)	CPU decision (ms)	CUDA decision (ms)	Params (MiB)	CUDA peak (MiB)
20/30/3	**3.9852**	4.0636	4.0337	4.0886	6.33	1.117	1.769	1.20	35.40
40/60/4	**4.2007**	4.2675	4.2558	4.3069	9.06	2.318	3.096	3.38	39.78
78/100/5	**4.5364**	4.5597	4.5426	4.5770	12.47	4.160	5.144	8.33	49.68

DRAPE has the lowest mean cost at all three sizes, but only three seeds are available. At full size, its mean advantages are 0.0233 over EC-MADRL, 0.0062 over OOA, and 0.0406 over uEDC-L; the OOA difference favors DRAPE in two seeds and its bootstrap interval crosses zero. EC-MADRL's training times are 5.97, 8.10, and 11.10 seconds, compared with 6.33, 9.06, and 12.47 seconds for DRAPE. At full size, the actor uses 380,224 parameters and all centralized critics use 1,803,525; the corresponding numbers are 90,642/222,723 at small scale and 195,376/691,204 at medium scale. A full CPU decision takes 4.160 ms, of which 0.737 ms is actor evaluation and 3.415 ms is projection. CUDA is slower for these single-state, small-batch calls because launch and transfer overhead dominate; no claim is made about large batched serving.

K Optimization and Subgroup Diagnostics

Over 60 training episodes, terminal-ten minus initial-ten cost is -0.284 on average for DRAPE and -0.226 for EC-MADRL. Four of five DRAPE seeds and three of five EC-MADRL seeds finish lower than they begin. The remaining nonmonotonicity is visible rather than hidden by averaging; RL optimization in this setting is noisy, and the unsmoothed held-out evaluation remains the basis for method comparisons.

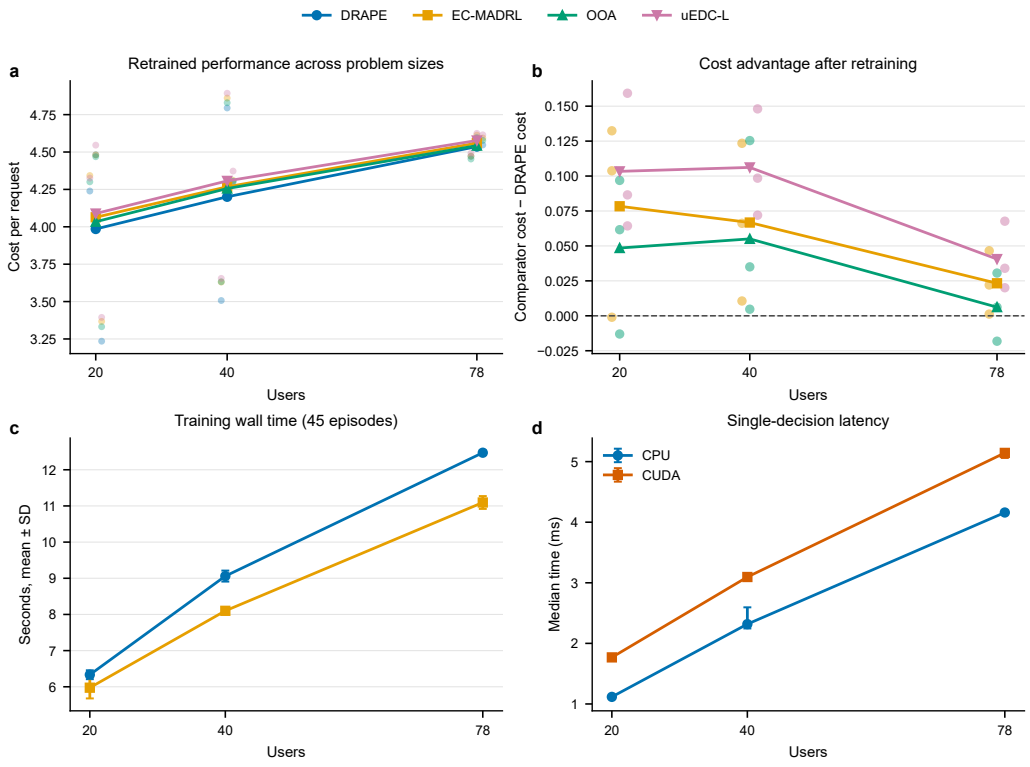

Figure 10: Retrained scaling and execution measurements. Panel (a) shows three seed-level cost points and their mean line for each method. Panel (b) shows three paired DRAPE-advantage points and their mean line. Panel (c) reports training wall time with SD error bars. Panel (d) reports median decision time with timing-block IQR bars.

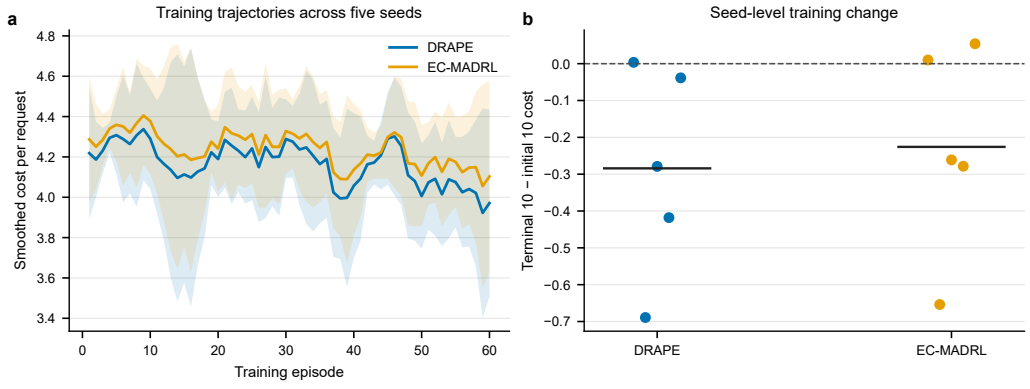

Figure 11: Optimization traces in the five-seed stress configuration. Curves are seven-episode moving averages; thin lines are individual seeds and thick lines are seed means. The initial and terminal ten-episode summaries are descriptive and are not treated as independent test samples.

RITE personality MAE across increasing activity quartiles is 0.1815, 0.1728, 0.1892, and 0.2052, with overlapping user-bootstrap intervals. Spearman correlation between log rating activity and user-level personality MAE is $\rho = 0.146$, $p = 0.201$ ($n = 78$), providing no resolved monotone relation. For response prediction, personality-conditioned versus context-only MAE is 0.1387/0.1434, 0.1316/0.1303, 0.1273/0.1253, and 0.1507/0.1437 across the same quartiles. The sign changes after the first quartile, again arguing against a general personality increment.

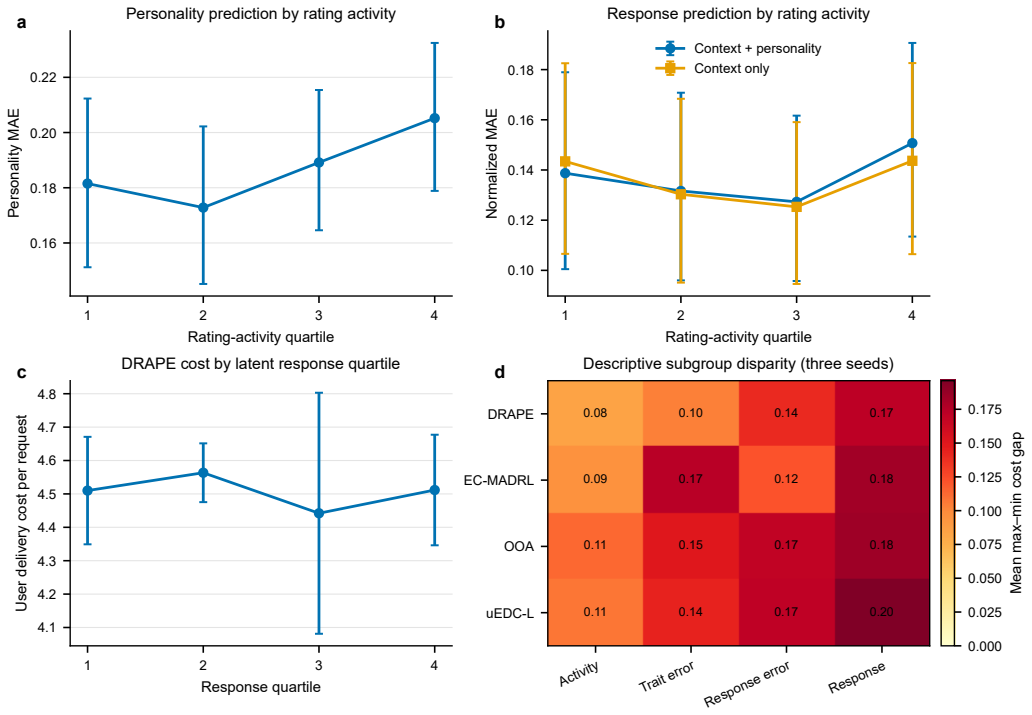

Figure 12: Prediction and downstream subgroup diagnostics. User groups are formed from rating activity, response level, personality-prediction error, or response-prediction error. Intervals for prediction errors bootstrap users; downstream intervals use three complete training seeds. These are stability checks, not protected-group fairness estimates.

Downstream per-user delivery costs are grouped by response quartile, response-error quartile, personality-error quartile, and activity tertile in the full 78-user scale experiment. DRAPE's response-quartile means are 4.510, 4.563, 4.442, and 4.512, with wide three-seed intervals and no monotone response disadvantage. Its mean max–min cost gaps are 0.1709, 0.1400, 0.1033, and 0.0828 for response, response error, personality error, and activity, respectively. The largest mean gap across the four methods is 0.1961 for uEDC-L by response quartile. Because these groups are data-derived, small, and not protected attributes, the analysis can reveal instability but cannot certify fairness.

L Reproducibility Record

The experiment package records user mappings, preprocessing metadata, fitted predictions, projection tuning, environment audits, run manifests, seed-level results, paired tests, timing blocks, and figure source data. The formal comparison can be reproduced from the prepared bundle with the fixed seeds:

```
python audit_fairness_v2.py
python run_comparison_v2.py --methods drape ec_madrl ooa \
  dra_sac cohcto uedc_l --seeds 2027 2028 2029 2030 2031 \
  2032 2033 2034 2035 2036 --output-dir results_n10
```

The full extension, audit, and figure regeneration use:

```
python run_extended_experiments.py \
  --sections mapping sensitivity scalability \
  efficiency formal prediction --device auto \
  --output-dir extended_results
python audit_extended_results.py
python plot_extended_results.py --results extended_results \
```

```
  --paper-figures ../../iclr2027_v2_work/figures
```

The delivered ten-seed directory combines two predeclared five-seed batches. The extended audit verifies 60/60 mapping rows, 640/640 stress rows, 36/36 scale rows, and 60/60 timing rows, as well as route sums, cache capacities, utilization bounds, and duplicate keys. Software versions are Python 3.8.20, NumPy 1.24.4, pandas 1.5.2, and PyTorch 2.4.1+cu118. CUDA measurements use an NVIDIA GeForce RTX 4060 Ti with 8 GB memory. Every new figure is exported as PDF, SVG, 300-dpi PNG, and 600-dpi LZW TIFF with its source CSV. An anonymous archival link should be added before submission.