ROBUST MIXTURE MODELS FOR ALGORITHMIC FAIRNESS UNDER LATENT HETEROGENEITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Machine learning models optimized for average performance can perform poorly on vulnerable subpopulations. Existing approaches often rely on predefined groups, yet fairness-relevant subgroup structure may be latent, intersectional, and driven by complex interactions among continuous and discrete attributes. We introduce **ROME** (**RO**bust **M**ixture **E**nsemble), a framework that learns latent group structure while optimizing worst-group predictive performance. ROME connects latent-variable modeling with distributionally robust optimization (DRO) through two complementary approaches: an Expectation-Maximization formulation with robust aggregation for linear models and a neural Mixture-of-Experts formulation for nonlinear settings. Across simulations and three real-world regression datasets, ROME improves worst-group performance while maintaining competitive overall accuracy against group-aware and group-label-free robust learning methods. ROME enables robust prediction when fairness-relevant attributes are available for subgroup discovery but restricted from direct use in group-specific outcome models.

1 INTRODUCTION

Machine learning models are increasingly used to support decisions in high-stakes domains such as healthcare, criminal justice, and finance. Strong average predictive performance, however, does not ensure reliable performance for all subpopulations: a model can achieve low overall error while performing substantially worse for smaller or systematically disadvantaged groups (Barocas & Selbst, 2016). Reducing such disparities is therefore important for developing reliable and fair predictive systems.

A central difficulty is that vulnerable subpopulations may not be known in advance. Even when sensitive attributes $\boldsymbol{S}$, such as sex, race, and socioeconomic characteristics, are available during model development, disparities may arise from unknown interactions among continuous and categorical attributes. Manually defining such groups can require arbitrary discretization and produce many small intersectional groups. Meanwhile, directly conditioning an outcome model on $\boldsymbol{S}$ may be undesirable or restricted by institutional, policy, or legal requirements. Thus, $\boldsymbol{S}$ may be valuable for identifying vulnerable subpopulations even when group-specific outcome models are restricted to non-sensitive predictive features $\boldsymbol{A}$.

1.1 RELATED WORK

A large literature studies fairness and robustness when sensitive groups are observed and specified in advance. For regression, Fair Regression (Agarwal et al., 2019) introduces bounded group loss constraints that control prediction loss within predefined groups, while GroupDRO (Sagawa et al., 2020) optimizes worst-group performance when group memberships are known. These approaches provide strong protection for known groups but require the groups over which robustness is enforced to be specified in advance. A complementary line of work addresses vulnerable subpopulations without predefined demographic group labels. ARL (Lahoti et al., 2020) adversarially reweights poorly performing observations, while GEORGE (Sohoni et al., 2020) and JTT (Liu et al., 2021) identify hidden or error-prone subpopulations from learned representations or prediction errors. The latter two target classification, and their subgroup identification mechanisms do not directly extend to continuous outcomes.

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.

Mixture models and mixture-of-experts provide classical and neural frameworks for representing population heterogeneity (McLachlan, 2000; Dempster et al., 1977; Jordan & Jacobs, 1994; Shazeer et al., 2017), while robust optimization provides complementary tools for protecting performance across heterogeneous populations (Sagawa et al., 2020; Wang et al., 2023). We consider a complementary setting where $\boldsymbol{S}$ is available for learning subgroup structure, while group-specific outcome models depend only on $\boldsymbol{A}$. We propose **ROME** (**RO**bust **M**ixture **E**nsemble), which combines latent subgroup discovery with distributionally robust optimization (DRO) to improve worst-group predictive performance.

1.2 Our Contributions

First, we study group-robust prediction under latent heterogeneity involving continuous and categorical fairness-relevant attributes. Second, we develop two formulations: ROME-EM combines mixture regression with robust aggregation for linear settings, while ROME-MoE jointly learns soft subgroup structure and optimizes average and worst-group performance in nonlinear settings. Third, we evaluate ROME through simulations and three regression datasets against five baselines, and examine robustness across external subgroup definitions and overall–worst-group performance trade-offs.

2 Problem Setup

We observe training data $\mathcal{D} = \{(\boldsymbol{A}_i, \boldsymbol{S}_i, Y_i)\}_{i=1}^n$, where $Y_i \in \mathbb{R}$ is a continuous outcome, $\boldsymbol{A}_i \in \mathbb{R}^{p_A}$ contains non-sensitive predictive features, and $\boldsymbol{S}_i \in \mathbb{R}^{p_S}$ contains fairness-relevant attributes. We consider a population with G latent subgroups, where $\boldsymbol{S}$ is available during development to inform subgroup structure but may be restricted from direct use in group-specific outcome models. Our goal is to learn this latent structure while improving worst-group performance without substantially sacrificing overall predictive accuracy.

2.1 Group-Robust Fairness Objective

We focus on disparities in predictive performance across subpopulations. Let $\mathcal{G}$ denote a collection of subpopulations and let $Z \in \mathcal{G}$ denote group membership. For a predictor f, define the group-specific risk

$$R_g(f) = \mathbb{E}[\ell\{Y, f(\boldsymbol{X})\} \mid Z = g], \qquad g \in \mathcal{G},$$

where ℓ is the squared loss for the continuous outcomes considered in this work. Our primary group-robust criterion is the worst-group risk

$$R_{\text{worst}}(f) = \max_{g \in \mathcal{G}} R_g(f).$$

Reducing $R_{\text{worst}}(f)$ protects against disproportionately poor predictive performance in the most adversely affected subgroup, while overall risk measures population-level predictive accuracy. We therefore consider both worst-group and overall performance when evaluating the trade-off between subgroup robustness and predictive accuracy.

Remark 1. Our group-robust objective differs from classical parity-based fairness criteria. Demographic parity requires predictions to be independent of sensitive attributes, while equalized odds requires this independence conditional on the outcome (Hardt et al., 2016). Extensions to continuous outcomes and sensitive attributes require different formulations (Agarwal et al., 2019; Narasimhan et al., 2020). Our objective instead targets disproportionately large predictive error across subpopulations and therefore does not imply satisfaction of other fairness criteria. Importantly, ROME's latent groups are distinct from the externally defined intersectional groups used only to audit performance in the real-data experiments.

3 ROME-EM For Linear Models

3.1 Latent Mixture Model

ROME-EM models population heterogeneity through G latent groups. Let $z_{ij} \in \{0, 1\}$ denote the latent group indicator where $z_{ij} = 1$ if observation i belongs to group j. The membership

probabilities follow a multinomial logistic model using selected sensitive attributes $\boldsymbol{S}_{i,\text{mem}} = \{S_{ik} : k \in \mathcal{I}_{\text{mem}}\}$

$$P(z_{ij} = 1 \mid \boldsymbol{S}_{i,\text{mem}}) = \frac{\exp(\gamma_j^\top \boldsymbol{S}_{i,\text{mem}})}{\sum_{k=1}^{G} \exp(\gamma_k^\top \boldsymbol{S}_{i,\text{mem}})} \tag{1}$$

where $\mathcal{I}_{\text{mem}} \subseteq \{1, \cdots, p_S\}$ denotes the indices of sensitive features used in the group membership model, and $\gamma_j \in \mathbb{R}^{|\mathcal{I}_{\text{mem}}|}$ are group-specific membership parameters. Given group membership, the continuous outcome follows a group-specific linear model:

$$Y_i \mid z_{ij} = 1 \sim \mathcal{N}(\boldsymbol{\omega}_j^\top \boldsymbol{X}_i, \sigma^2) \tag{2}$$

with $\boldsymbol{X}_i = [1, \boldsymbol{A}_i, \boldsymbol{S}_{i,\text{out}}]$ being the design vector, $\boldsymbol{S}_{i,\text{out}} = \{S_{ik} : k \in \mathcal{I}_{\text{out}}\}$ the sensitive features included in the outcome prediction model and $\boldsymbol{\omega}_j \in \mathbb{R}^{1+p_A+|\mathcal{I}_{\text{out}}|}$. The feature selection indices $\mathcal{I}_{\text{mem}}$ and $\mathcal{I}_{\text{out}}$ enable flexible modeling based on domain constraints. For instance, we might use all demographic features to identify groups while excluding protected attributes from outcome prediction. Specifically, we distinguish two roles of sensitive attributes: $\boldsymbol{S}_{i,\text{mem}}$ contains variables allowed to inform the estimation of latent subgroup structure, while $\boldsymbol{S}_{i,\text{out}}$ contains variables permitted to enter the group-specific outcome model[1].

3.2 EM Algorithm

Since group memberships are latent, we use the expectation maximization (EM) algorithm to estimate parameters $\boldsymbol{\Theta} = \{\gamma_1, \ldots, \gamma_G, \boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_G\}$.

Initialization. The initial hard assignments $\{z_{ij}^{(0)}\}_{i=1}^n$ provide a warm start for EM and are replaced by soft probabilities in subsequent iterations. They can either be (1) specified based on domain knowledge, for example by clustering sensitive attributes to obtain initial hard assignments, or (2) randomly sampled from the discrete uniform distribution on $\{1, \ldots, G\}$. For membership parameters γ_j, we fit a logistic regression of the initial group indicators on $\boldsymbol{S}_{\text{mem}}$. For outcome parameters $\boldsymbol{\omega}_j$, we fit group-specific linear regressions when n_j is sufficiently large (Ng et al., 2011; Levine & Casella, 2001); otherwise, we use pooled regression estimates to ensure numerical stability.

E-Step. Given current parameters $\{\gamma_j^{(t)}, \boldsymbol{\omega}_j^{(t)}\}_{j=1}^G$, we compute the posterior probabilities:

$$w_{ij}^{(t)} = \frac{p_{ij}^{(t)} \cdot \ell_{ij}^{(t)}}{\sum_{k=1}^{G} p_{ik}^{(t)} \cdot \ell_{ik}^{(t)}},$$

where $p_{ij}^{(t)} = P(z_{ij} = 1 \mid \boldsymbol{S}_{i,\text{mem}}; \gamma_j^{(t)})$ is the membership probability of observation i under group j and

$$\ell_{ij}^{(t)} = \exp\left(-\frac{1}{2}(Y_i - \boldsymbol{\omega}_j^{(t)\top} \boldsymbol{X}_i)^2\right)$$

is the Gaussian likelihood, with fixed variance absorbed into the proportionality.

M-Step with Backtracking Line Search. To promote monotonic increase in the log-likelihood, we employ backtracking line search (Liang, 2021) for parameter updates. For each parameter update from $\boldsymbol{\theta}^{(t)}$ to $\boldsymbol{\theta}^{(t+1)}$, we set: $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + \alpha(\boldsymbol{\theta}_{\text{new}} - \boldsymbol{\theta}^{(t)})$ starting with $\alpha = 0.5$ and halving α until the likelihood improves or $\alpha < \tau_2$. For each group, we update γ_j via weighted logistic regression with weights $w_{ij}^{(t)}$, using a quasi-binomial family to handle potential overdispersion:

$$\gamma_j^{\text{new}} = \arg\max_{\gamma} \sum_{i=1}^{n} w_{ij}^{(t)} \left[\gamma^\top \boldsymbol{S}_{i,\text{mem}} - \log\left(\sum_{k=1}^{G} \exp(\gamma_k^\top \boldsymbol{S}_{i,\text{mem}})\right)\right]$$

We solve the weighted least squares problem: $\boldsymbol{\omega}_j^{\text{new}} = (\boldsymbol{X}^\top \boldsymbol{W}_j^{(t)} \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{W}_j^{(t)} \boldsymbol{Y}$, where $\boldsymbol{W}_j^{(t)} = \text{diag}(w_{1j}^{(t)}, \ldots, w_{nj}^{(t)})$.

[1]The determination of which attributes belong to $\boldsymbol{S}_{i,\text{mem}}$ and $\boldsymbol{S}_{i,\text{out}}$ is entirely guided by domain knowledge, institutional guidelines, and fairness/compliance requirements, not by the method itself.

Convergence and Model Selection. The algorithm terminates when the total parameter change falls below tolerance τ_1:

$$\sum_{j=1}^{G} \left(||\gamma_j^{(t+1)} - \gamma_j^{(t)}||_1 + ||\boldsymbol{\omega}_j^{(t+1)} - \boldsymbol{\omega}_j^{(t)}||_1 \right) < \tau_1$$

or the maximum iterations are reached. As with finite mixture models generally, convergence need not correspond to a global optimum and may depend on initialization. In practice, the number of groups G can be selected using information criteria such as AIC or BIC computed from the maximized log-likelihood.

3.3 DRO Aggregation

After obtaining group-specific models $\{\hat{\boldsymbol{\omega}}_1, \ldots, \hat{\boldsymbol{\omega}}_G\}$ from the EM, we treat the estimated latent components as source populations and apply the distributionally robust optimization (DRO) framework to construct a robust predictor that performs well across different mixture distributions of the latent groups.

DRO Formulation. We consider mixture distributions of the G group-specific conditionals:

$$\mathbb{P}^q_{Y|X} = \sum_{j=1}^{G} q_j \, \mathbb{P}^{(j)}_{Y|X}, \qquad q \in \mathcal{H},$$

where $\mathcal{H} = \left\{ \mathbf{v} \in \Delta^G : \|\mathbf{v} - \mathbf{v}_0\|_2 \leq c\sqrt{G} \right\} \subseteq \Delta^G$ is an uncertainty set over mixture weights. Here $\mathbf{v}_0$ represents a baseline mixture weight, taken by default as the uniform vector $(1/G, \ldots, 1/G)$, and $c \in [0, 1]$ controls the size of the uncertainty region. The distributionally robust predictor solves

$$f^* = \arg\max_{f \in \mathcal{F}} \min_{q \in \mathcal{H}} R_q(f)$$

where $R_q(f)$ denotes the explained variance under mixture weights q.

Closed-Form Solution: Following key results from Wang et al. (2023), when $\mathcal{F}$ consists of convex combinations of the group-specific estimated predictors $\hat{f}_j(\boldsymbol{X}) = \hat{\boldsymbol{\omega}}_j^\top \boldsymbol{X}$, the robust predictor $f_{\boldsymbol{v}}$ has the closed form:

$$f_{\boldsymbol{v}}(\boldsymbol{X}_i) = \sum_{j=1}^{G} v_j \cdot \hat{\boldsymbol{\omega}}_j^\top \boldsymbol{X}_i \tag{3}$$

where the optimal weights solve:

$$\boldsymbol{v}^* = \arg\min_{v \in \mathcal{H}} \boldsymbol{v}^\top \hat{\gamma} \boldsymbol{v} \tag{4}$$

Here γ is a $G \times G$ matrix, representing the empirical covariance of predictions across groups, with entries $\gamma_{k,j} = \mathbb{E}_{Q_X}[f^{(k)}(X) f^{(j)}(X)]$ and we compute $\hat{\gamma}$ by $\hat{\gamma}_{jk} = \frac{1}{n} \sum_{i=1}^{n} \hat{f}_j(\boldsymbol{X}_i) \cdot \hat{f}_k(\boldsymbol{X}_i)$.

Final Robust Predictor. Finally, the robust predictor of ROME-EM for $\boldsymbol{X}_{\text{new}}$ is:

$$\hat{Y}_{\text{ROME-EM}} = \sum_{j=1}^{G} v_j^* \cdot \hat{\boldsymbol{\omega}}_j^\top \boldsymbol{X}_{\text{new}} \tag{5}$$

ROME-EM makes explicit the connection between latent mixture estimation and robust aggregation. This linear construction provides the methodological motivation for extending latent-group robustness to the nonlinear mixture-of-experts setting developed next.

4 ROME-MoE for Nonlinear Models

ROME-EM relies on a linear multinomial membership model and may be misspecified when latent group structure involves nonlinear or complex feature interactions. Although neural mixture

Algorithm 1 ROME-MoE

Require: Training data $\mathcal{D} = \{(\boldsymbol{A}_i, \boldsymbol{S}_i, Y_i)\}_{i=1}^n$, number of experts G, DRO parameter $\alpha \in [0, 1]$, learning rate η, batch size B, epochs E
Ensure: Trained ROME-MoE model with parameters $\Theta = \{\theta_{\text{gate}}, \theta_1, \ldots, \theta_G\}$
Initialize: Gating network $g_{\theta_{\text{gate}}}$, Expert networks $\{f_{\theta_1}, \ldots, f_{\theta_G}\}$
for epoch $= 1$ to E **do**
  for each batch $\mathcal{B} \subset \mathcal{D}$ with $|\mathcal{B}| = B$ **do**
    for each $(\boldsymbol{A}_i, \boldsymbol{S}_i, Y_i) \in \mathcal{B}$ **do**
      if `ROME-MoE-S` **then**
        $\boldsymbol{w}_i = \text{Softmax}(g_{\theta_{\text{gate}}}(\boldsymbol{S}_i))$ ▷ `ROME-MoE-S`
      else ▷ `ROME-MoE-AS`
        $\boldsymbol{w}_i = \text{Softmax}(g_{\theta_{\text{gate}}}([\boldsymbol{A}_i; \boldsymbol{S}_i]))$
      end if
      $\hat{Y}_i = \sum_{j=1}^G w_{ij} \cdot f_{\theta_j}(\boldsymbol{A}_i)$ ▷ Weighted expert predictions
    end for
    for $j = 1$ to G **do**
      $\mathcal{I}_j = \{i \in \mathcal{B} : w_{ij} > 0.1\}$ ▷ Samples with significant membership
      if $|\mathcal{I}_j| > 0$ **then**
        $\mathcal{L}_j = \frac{1}{|\mathcal{I}_j|} \sum_{i \in \mathcal{I}_j} w_{ij} \cdot (Y_i - \hat{Y}_i)^2$
      else
        $\mathcal{L}_j = 0$
      end if
    end for
    // DRO Objective
    $\mathcal{L}_{\text{avg}} = \frac{1}{B} \sum_{i \in \mathcal{B}} (Y_i - \hat{Y}_i)^2$ ▷ Average loss
    $\mathcal{L}_{\text{worst}} = \max_{j \in \{1,\ldots,G\}} \mathcal{L}_j$ ▷ Worst group loss
    $\mathcal{L}_{\text{total}} = (1 - \alpha) \cdot \mathcal{L}_{\text{avg}} + \alpha \cdot \mathcal{L}_{\text{worst}}$
    // Backward Pass
    Compute gradients: $\nabla_\Theta \mathcal{L}_{\text{total}}$
    Update parameters: $\Theta \leftarrow \Theta - \eta \cdot \nabla_\Theta \mathcal{L}_{\text{total}}$
  end for
end for
return Trained model with parameters Θ

models can accommodate such structure (Nagpal et al., 2022), they do not retain ROME-EM's simple closed-form robust aggregation. We therefore extend ROME using Mixture of Experts (MoE) (Shazeer et al., 2017), with DRO directly integrated into training to balance average and worst-group performance. ROME-MoE treats experts as group-specific predictors and optimizes the robust objective end-to-end rather than applying a separate aggregation step.

Our ROME-MoE architecture consists of two key components: (1) **Gating network:** A neural network $g : \mathbb{R}^p \to \mathbb{R}^G$ maps input features to G logits, which are transformed by a softmax to obtain a probability distribution over the experts and determine soft group assignments. (2) **Expert network:** A collection of G neural networks $\{f_1, \ldots, f_G\}$ where each $f_j : \mathbb{R}^{p_A} \to \mathbb{R}$ specializes in predictions for its corresponding latent group using only non-sensitive features. We present two variants: ROME-MoE-S uses only sensitive features $\boldsymbol{S}$ for the gating network (assuming group membership is primarily determined by $\boldsymbol{S}$), while ROME-MoE-AS uses all features. The key architectural restriction in ROME-MoE is that $\boldsymbol{S}$ can influence group assignment through the gating network but is not provided directly to the expert prediction functions, which receive only $\boldsymbol{A}$.

Use of sensitive attributes. Sensitive features $\boldsymbol{S}$ are used by the gating network to inform latent subgroup structure, while the expert outcome models receive only $\boldsymbol{A}$. Thus, $\boldsymbol{S}$ can influence the final prediction through the gating weights but does not enter the expert prediction functions directly. Users determine which attributes are treated as sensitive according to the domain and relevant institutional, policy, or legal requirements.

The detailed implementation is available in Algorithm 1. Unlike Vanilla MoE training that minimizes average loss, ROME-MoE incorporates distributionally robust optimization directly into the training objective. Given a batch of data, we compute group assignments through the gating network and optimize:

$$\mathcal{L}_{\text{ROME-MoE}} = (1-\alpha)\cdot\mathcal{L}_{\text{avg}} + \alpha\cdot\mathcal{L}_{\text{worst}} \tag{6}$$

where $\mathcal{L}_{\text{avg}}$ is the average loss across all samples, $\mathcal{L}_{\text{worst}} = \max_{j\in\{1,\ldots,G\}}\mathcal{L}_j$ is the worst group loss, and $\alpha \in [0,1]$ controls the trade-off between average and worst-case performance. For each latent group, $\mathcal{L}_j$ is computed using observations with membership weight $w_{ij} > 0.1$, with their squared errors weighted by w_{ij}, as detailed in Algorithm 1. In practice, we found $\alpha \in [0.05, 0.1]$ provides a good balance between average and worst-group performance.

5 Experiments

5.1 Simulation Study

Data Generation and Setup. We conducted a comprehensive simulation study to evaluate ROME-EM's ability to recover group-specific parameters and improve worst-group performance. We generated data with $n = 8,000$ observations from $G = 4$ latent groups, each with distinct parameter vectors and mixing proportions. Group membership was determined by a multinomial logistic model based on sensitive attributes $\boldsymbol{S} \in \mathbb{R}^5$, with each group having distinct membership parameters. For the outcome model, we generated $p = 20$ total features comprising $p_A = 15$ non-sensitive features and $p_S = 5$ sensitive attributes.

The true regression coefficients β_j for each group were constructed using a scaled perturbation approach to introduce meaningful heterogeneity across groups while maintaining realistic parameter magnitudes. Complete simulation parameters are provided in Appendix A, with additional details in Appendix A.1. To assess robustness to initialization, we intentionally misspecified the starting conditions by randomly reassigning 50% of observations to incorrect groups before running ROME-EM.

For the DRO optimization stage, we evaluated multiple constraint values to examine the trade-off between overall and worst-group predictive performance (detailed in Appendix A). Each of the 100 simulation replications was evaluated on an independent test set of $n = 8,000$ observations generated from the same underlying distributions.

Results. Across 100 simulation replications, ROME-EM demonstrated superior parameter recovery and worst-group performance compared to pooled regression. Figure 1 illustrates parameter estimation accuracy for a representative latent group: ROME-EM estimates (blue) cluster near the true values (red dashed lines), whereas pooled estimates (coral) exhibit substantial bias. Results for the other three latent groups in Appendix A.2 show similar patterns. The worst-group MSE decreased from a mean of 36.56 under pooled regression to 32.59 with ROME-EM (optimal constraint), an approximately 11% reduction (paired t-test: $p < 0.001$); the corresponding distribution across simulations is shown in Appendix A.2.

5.2 Real-Data Analysis

We evaluated ROME on three real-world datasets from diverse domains: the Law School Admissions dataset (Wightman, 1999), the Communities & Crime dataset (Redmond, 2002), and the American Community Survey Public Use Microdata Sample (ACS PUMS) (Bureau, 2003), accessed via the Folktables library (Ding et al., 2021).

For each dataset, we compared seven models: (1) Baseline MLP, a standard MLP using both $\boldsymbol{A}$ and $\boldsymbol{S}$ as predictive features; (2) Baseline MLP-Fair, an MLP using only non-sensitive features $\boldsymbol{A}$; (3) Adversarially Reweighted Learning (ARL) (Lahoti et al., 2020), which adversarially reweights poorly performing observations without requiring predefined demographic groups and is adapted here to continuous outcomes using squared error; (4) Fair Regression-BGL (Agarwal et al., 2019), which minimizes overall prediction loss subject to bounded loss within predefined sensitive groups; (5) Vanilla MoE, which uses both $\boldsymbol{A}$ and $\boldsymbol{S}$ for gating and expert prediction; (6) ROME-MoE-S, which uses $\boldsymbol{S}$ for latent-group gating and $\boldsymbol{A}$ for expert prediction; and (7) ROME-MoE-AS, which

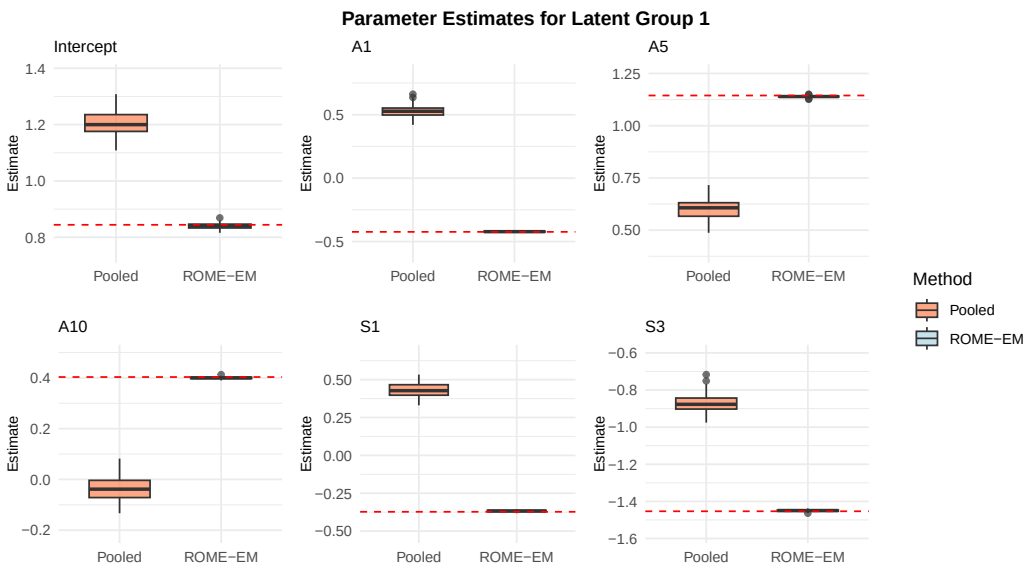

Figure 1: Box plots comparing parameter recovery by ROME-EM and pooled regression for latent group 1 (over 100 simulations). Red dashed lines denote ground truth values for each parameter.

uses both $\boldsymbol{A}$ and $\boldsymbol{S}$ for latent-group gating while restricting expert prediction to $\boldsymbol{A}$. Under our architectural criterion, models (2)–(4), (6), and (7) are considered fair because sensitive attributes are not provided as direct inputs to their outcome prediction models.

We set the DRO parameter $\alpha = 0.05$ for all main ROME-MoE experiments and examine its effect in the ablation study in Section 5.2. We report both MSE and R^2 for overall test-set performance and for the worst-performing evaluation subgroup. Because ground-truth latent groups are unavailable in the real datasets, worst-group performance is audited using externally defined subgroups based on sensitive attributes. The main evaluation uses (S_2, S_3) with quartile discretization for Law School, (S_2, S_3) with median discretization for Communities & Crime, and S_2 for ACS PUMS. Additional subgroup definitions are examined in the ablation study.

Table 1 summarizes the main results. ROME-MoE achieves strong worst-group performance while retaining competitive overall predictive accuracy. For worst-group MSE, both ROME variants significantly outperform the best-performing fair baseline across datasets. ROME-MoE-AS attains the lowest worst-group MSE among fair models on Law School and Communities & Crime, while ROME-MoE-S attains the lowest worst-group MSE on ACS. At the same time, the overall MSE of the best-performing ROME variant on each dataset is not significantly different from that of the best-performing baseline among all models.

The R^2 results provide a complementary view. On Law School and ACS PUMS, ROME achieves the highest worst-group R^2 among fair models, with significant improvements over the best-performing fair baselines. On Communities & Crime, Fair Regression-BGL attains the highest mean worst-group R^2 at 0.5361, while ROME-MoE-AS achieves a nearly identical value of 0.5356 with no significant difference between the two methods. ROME-MoE-AS nevertheless achieves the highest overall R^2 among the fair models on this dataset. Together with the MSE results, these findings indicate that ROME improves worst-group prediction error while generally preserving strong overall predictive performance.

ROME-EM performed well in the simulation study but encountered difficulties on these real datasets. Without informative initialization or sufficiently strong mixture structure, the algorithm sometimes converged to solutions with similar parameters across groups, effectively approaching pooled regression, or failed to converge. These results suggest that the parametric mixture assumptions of ROME-EM can be restrictive for complex real-world heterogeneity. In contrast, ROME-MoE learns the latent structure and predictive models jointly through the gating and expert networks and was more effective in these real-data settings.

Table 1: Overall and worst-group predictive performance on the test sets, averaged over 10 random seeds (mean with standard error in parentheses). Best-performing fair models are shown in **bold**. For worst-group metrics, each ROME variant is compared with the best-performing fair baseline; for overall metrics, each ROME variant is compared with the best-performing baseline among all models. Statistical significance is assessed using paired two-sided t-tests across the 10 matched random seeds: $^{*}p < 0.05$, $^{**}p < 0.01$, and $^{***}p < 0.001$; ^{ns} indicates no significant difference ($p > 0.05$).

			Overall		Worst-Group	
Dataset	**Model**	**Fair**	**MSE**	$\boldsymbol{R^2}$	**MSE**	$\boldsymbol{R^2}$
Law School Admissions Council	Baseline MLP		0.7196 (0.0006)	0.1776 (0.0007)	0.8184 (0.0027)	0.1225 (0.0006)
	Baseline MLP - Fair	✓	0.7368 (0.0005)	0.1579 (0.0005)	0.8300 (0.0012)	0.0933 (0.0015)
	ARL	✓	0.7355 (0.0004)	0.1594 (0.0004)	0.8287 (0.0021)	0.0929 (0.0017)
	FairReg-BGL	✓	0.7352 (0.0005)	0.1597 (0.0005)	0.8269 (0.0017)	0.0928 (0.0008)
	Vanilla MoE		0.7209 (0.0007)	0.1761 (0.0008)	0.8233 (0.0018)	0.1210 (0.0012)
	ROME-MoE-S	✓	0.7217 (0.0014)^{ns}	0.1752 (0.0016)^{ns}	0.8144 (0.0011)***	0.1148 (0.0024)***
	ROME-MoE-AS	✓	**0.7200 (0.0015)**^{ns}	**0.1772 (0.0017)**^{ns}	**0.8127 (0.0023)*****	**0.1181 (0.0021)*****
Communities and Crime	Baseline MLP		0.0205 (0.0002)	0.6149 (0.0029)	0.0285 (0.0005)	0.5274 (0.0025)
	Baseline MLP - Fair	✓	0.0235 (0.0002)	0.5570 (0.0032)	0.0326 (0.0004)	0.4882 (0.0062)
	ARL	✓	0.0225 (0.0001)	0.5751 (0.0019)	0.0300 (0.0003)	0.5276 (0.0045)
	FairReg-BGL	✓	0.0223 (0.0001)	0.5793 (0.0016)	0.0298 (0.0002)	**0.5361 (0.0018)**
	Vanilla MoE		0.0207 (0.0002)	0.6105 (0.0033)	0.0286 (0.0005)	0.5281 (0.0030)
	ROME-MoE-S	✓	0.0207 (0.0002)^{ns}	0.6095 (0.0032)^{ns}	0.0287 (0.0004)**	0.5270 (0.0031)
	ROME-MoE-AS	✓	**0.0204 (0.0001)**^{ns}	**0.6166 (0.0024)**^{ns}	**0.0271 (0.0002)*****	0.5356 (0.0036)^{ns}
American Community Survey Public Use Microdata Sample	Baseline MLP		0.0044 (0.0000)	0.5463 (0.0035)	0.0046 (0.0000)	0.4907 (0.0047)
	Baseline MLP - Fair	✓	0.0053 (0.0000)	0.4554 (0.0024)	0.0053 (0.0000)	0.4237 (0.0031)
	ARL	✓	0.0052 (0.0000)	0.4571 (0.0020)	0.0053 (0.0000)	0.4245 (0.0036)
	FairReg-BGL	✓	0.0053 (0.0000)	0.4543 (0.0032)	0.0053 (0.0000)	0.4272 (0.0024)
	Vanilla MoE		0.0047 (0.0000)	0.5104 (0.0034)	0.0048 (0.0000)	0.4663 (0.0036)
	ROME-MoE-S	✓	**0.0045 (0.0001)**^{ns}	**0.5299 (0.0078)**^{ns}	**0.0047 (0.0001)*****	**0.4779 (0.0093)****
	ROME-MoE-AS	✓	0.0047 (0.0001)	0.5170 (0.0054)	0.0049 (0.0000)***	0.4554 (0.0053)**

The main table 1 reports one representative evaluation subgroup scheme for each dataset. We additionally evaluate two alternative subgroup definitions per dataset in Section 5.2, with full results provided in Appendix C.2. Experimental details, including data preprocessing and hyperparameter selection, are provided in Appendices B and D.2.

Ablation Studies. We conducted two ablation studies to examine key design choices in ROME.

(1). Impact of DRO Parameter α. We varied the DRO parameter α to examine the contribution of the robust training objective beyond the MoE architecture alone. As shown in Figures 7, 2, and 8, increasing α generally improves worst-group performance, with the gains tending to saturate at modest values of α, while overall MSE remains relatively stable across a broad range of values. These results support our choice of $\alpha = 0.05$ for the main experiments, which places substantially greater weight on overall predictive accuracy while still incorporating worst-group robustness. The broadly similar patterns across datasets and both ROME-MoE variants support the contribution of the DRO objective beyond the MoE architecture alone.

(2). Impact of Evaluation Subgroups. To assess whether the observed improvements depend on a particular evaluation partition, we evaluated the models under multiple subgroup definitions based on different combinations of sensitive attributes. Categorical attributes retain their natural categories, while continuous attributes are partitioned using median or quartile cut points. Results for the alternative evaluation schemes are reported in Appendix C.2. Across these subgroup definitions, ROME generally maintains strong worst-group performance, indicating that the main findings are not driven by a single choice of evaluation partition.

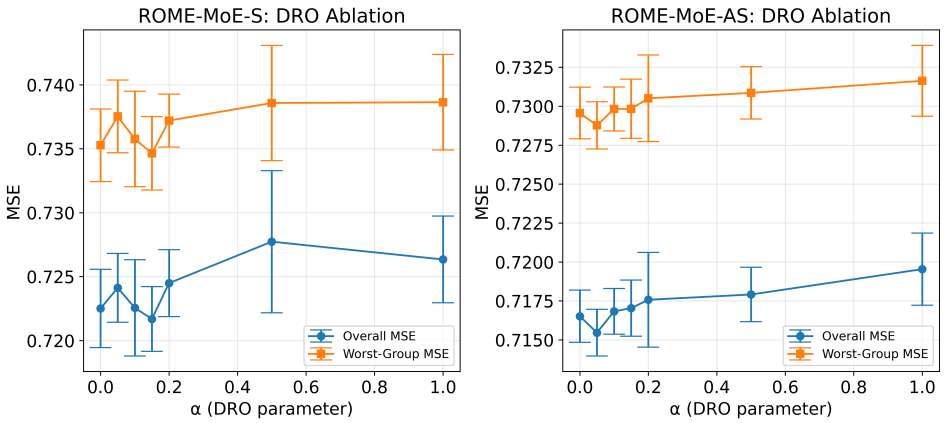

Figure 2: Effect of the DRO parameter α on ROME-MoE performance for the Law dataset over 10 seeds.

6 Discussion

ROME addresses group-robust prediction when fairness-relevant attributes are observed but the subgroup structure underlying predictive disparities is unknown. ROME-EM connects latent mixture modeling with robust aggregation and performs well under correctly specified mixture structure, as demonstrated in our simulations. However, it was sensitive to initialization and parametric assumptions on the real datasets. ROME-MoE provides a more flexible nonlinear alternative and achieved strong worst-group performance while maintaining competitive overall predictive performance.

A key feature of ROME is the separation between subgroup modeling and group-specific outcome prediction. Rather than specifying intersectional groups in advance, ROME uses fairness-relevant attributes to learn a soft representation of population heterogeneity. In ROME-MoE, sensitive attributes may inform the gating network, while expert outcome models receive only non-sensitive features. The learned latent groups need not correspond to demographic categories; performance can instead be audited using externally defined subgroups. This allows latent structure to guide robust model fitting without treating a predefined partition as the ground-truth source of predictive disparity.

ROME requires fairness-relevant attributes during model development and is therefore limited when such attributes cannot be collected or observed. ROME-EM offers a structured, interpretable representation but relies on parametric assumptions and initialization, whereas ROME-MoE captures more complex nonlinear structure at the cost of interpretability. This may limit ROME-MoE in applications requiring transparent explanations of subgroup assignment or individual predictions. Developing interpretable nonlinear formulations and explanation methods for latent-group robust prediction is an important direction for future work.

Reproducibility Statement

The simulation design, data-generating process, real-data construction, and evaluation protocols are described in Section 5.1 and the appendices. The ROME-MoE training procedure is provided in Algorithm 1, with neural network architectures, hyperparameter selection, and additional experimental details reported in the appendices. All datasets used in this study are publicly available. Anonymous source code is available at `https://anonymous.4open.science/r/ROME-7CA4`.

Ethics Statement

This work uses publicly available datasets and does not involve new data collection or human-subject recruitment. The study focuses on group-robust prediction using fairness-relevant attributes. Sensitive attributes are used to inform subgroup modeling, while their direct use in group-specific

outcome models may be restricted depending on the modeling setting. The appropriateness of using such attributes should be assessed according to the legal, institutional, and ethical requirements of the intended application.

ACKNOWLEDGMENTS

Acknowledgments are withheld for anonymous review.

AI USE STATEMENT

We used generative AI tools (ChatGPT) for editing and proofreading manuscript text. All AI-assisted output was reviewed and revised by the authors. We take responsibility for the final content of this work.

REFERENCES

Alekh Agarwal, Miroslav Dudik, and Zhiwei Steven Wu. Fair regression: Quantitative definitions and reduction-based algorithms. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 120–129. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/agarwal19d.html.

Solon Barocas and Andrew D Selbst. Big data's disparate impact. *Calif. L. Rev.*, 104:671, 2016.

US Census Bureau. American community survey, public use microdata sample, 2003.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1): 1–22, 1977.

Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. Retiring adult: New datasets for fair machine learning, 2021.

Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016.

Michael I Jordan and Robert A Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2):181–214, 1994.

Preethi Lahoti, Alex Beutel, Jilin Chen, Kang Lee, Flavien Prost, Nithum Thain, Xuezhi Wang, and Ed H. Chi. Fairness without demographics through adversarially reweighted learning. *CoRR*, abs/2006.13114, 2020. URL https://arxiv.org/abs/2006.13114.

Richard A Levine and George Casella. Implementations of the monte carlo em algorithm. *Journal of Computational and Graphical Statistics*, 10(3):422–439, 2001.

Jingchen Liang. Gradient descent and newton's method with backtracking line search in linear regression. In *2021 2nd International Conference on Computing and Data Science (CDS)*, pp. 394–397. IEEE, 2021.

Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *International conference on machine learning*, pp. 6781–6792. PMLR, 2021.

Geoffrey McLachlan. Finite mixture models. *A wiley-interscience publication*, 2000.

Chirag Nagpal, Steve Yadlowsky, Negar Rostamzadeh, and Katherine Heller. Deep cox mixtures for survival regression, 2022. URL https://arxiv.org/abs/2101.06536.

Harikrishna Narasimhan, Andrew Cotter, Maya Gupta, and Serena Wang. Pairwise fairness for ranking and regression. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 5248–5255, 2020.

Shu Kay Ng, Thriyambakam Krishnan, and Geoffrey J McLachlan. The em algorithm. In *Handbook of computational statistics: concepts and methods*, pp. 139–172. Springer, 2011.

Michael Redmond. Communities and Crime. UCI Machine Learning Repository, 2002. DOI: https://doi.org/10.24432/C53W3X.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization, 2020. URL https://arxiv.org/abs/1911.08731.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Nimit Sohoni, Jared Dunnmon, Geoffrey Angus, Albert Gu, and Christopher Ré. No subclass left behind: Fine-grained robustness in coarse-grained classification problems. *Advances in Neural Information Processing Systems*, 33:19339–19352, 2020.

Zhenyu Wang, Peter Bühlmann, and Zijian Guo. Distributionally robust machine learning with multi-source data. *arXiv preprint arXiv:2309.02211*, 2023.

Linda F. Wightman. Law school admission council national longitudinal bar passage study (LSBC), 1999. URL https://doi.org/10.3886/ICPSR22141.v1.

A Simulation Study: Additional Details

A.1 Parameter Specifications

We follow equation 2 to generate the data, using the model parameters as detailed in Section 5.1. Specifically, we simulate features $X = (A, S) \in \mathbb{R}^p$ by drawing $X \sim \mathcal{N}(0, \Sigma_X)$. Outcomes are generated by group-specific linear models: $Y_i = \beta_{G_i}^\top [1, A_i, S_i^{\text{out}}] + \varepsilon_i$ with $\varepsilon_i \sim \mathcal{N}(0, 0.1^2)$. To model mixture shift, we optionally use distinct membership parameters Γ (train) and Γ'(test). Finally, to mimic imperfect initialization, each true group label is flipped to a different class with small probability r to form G_0^{fake}.

Full details and implementations are provided in script 'demo.R' of the code repository.

Group Membership Parameters (γ): The membership parameter matrix $\gamma \in \mathbb{R}^{4\times 5}$ was specified as:

$$\gamma = \begin{bmatrix} 2.0 & 2.0 & 2.0 & 2.0 & 2.0 \\ -3.0 & -2.0 & -5.0 & 0.1 & 0.1 \\ 0.1 & -10.0 & 0.1 & 0.1 & 0.1 \\ -2.0 & -2.0 & -2.0 & -2.0 & -2.0 \end{bmatrix} \tag{7}$$

Outcome Model Coefficients (β): The true regression coefficients were generated using a heterogeneous scaling approach. Starting with a base vector $\boldsymbol{\beta}_0 = \mathbf{1}_{20}$, group-specific coefficients were created as $\boldsymbol{\beta}_j = \epsilon_j \odot \boldsymbol{\beta}_0$, where ϵ_j are scaling factors generated with heterogeneity parameter $\delta = 0.8$. Group intercepts were drawn uniformly from $[-1, 1]$. The final coefficient matrix $\boldsymbol{\beta}^T$

(transposed for space) is:

$$\boldsymbol{\beta}^T = \begin{bmatrix} 0.844 & 0.090 & 0.962 & 0.618 \\ -0.423 & 0.749 & 1.309 & 0.307 \\ 0.696 & -0.545 & 0.559 & 1.703 \\ -0.449 & 1.646 & -1.165 & 1.361 \\ -0.737 & -0.429 & 0.255 & 1.384 \\ 1.144 & -1.003 & 1.014 & -1.377 \\ 0.988 & 1.666 & -1.336 & -1.209 \\ -1.702 & -1.681 & -1.295 & 0.745 \\ 1.217 & 1.800 & -1.767 & 0.218 \\ -0.922 & -1.566 & 0.744 & -0.287 \\ 0.403 & -1.523 & 0.395 & -1.727 \\ 1.729 & 1.151 & 0.292 & 0.830 \\ -1.661 & 1.461 & 1.628 & -1.368 \\ -0.217 & 1.637 & 0.840 & -1.781 \\ -0.437 & -0.831 & -0.509 & 1.747 \\ -1.520 & -0.487 & -0.325 & -1.428 \\ -0.372 & -0.557 & 1.058 & 0.414 \\ 0.826 & -1.509 & -0.450 & 0.903 \\ -1.453 & -0.848 & -0.748 & 1.429 \\ -0.773 & 0.996 & 0.765 & 1.460 \\ -1.134 & -1.441 & 0.509 & -1.790 \end{bmatrix} \tag{8}$$

where rows 1-21 correspond to the intercept, A_1-A_{15}, and S_1-S_5 respectively, and columns 1-4 correspond to latent groups 1-4.

Algorithm Hyperparameters:

- EM convergence tolerance: $\tau_1 = 10^{-3}$ (parameter change), $\tau_2 = 5 \times 10^{-3}$ (line search)
- Maximum EM iterations: 100
- DRO constraint values: $\mathcal{C} \in \{1.0, 0.6, 0.5, 0.49, 0.48, 0.47, \ldots, 0.04, 0.03, 0.02\}$ (27 values total)
- Misspecification rate: 50% (proportion of observations with incorrect initial group assignment)

Data Structure: All features were generated from independent standard normal distributions ($\Sigma_X = \boldsymbol{I}_{20}$). The first 15 features served as non-sensitive features ($\boldsymbol{A}$), while the remaining 5 were treated as sensitive attributes ($\boldsymbol{S}$). All five sensitive attributes were included in both the membership ($\mathcal{I}_{\text{mem}} = \{1, 2, 3, 4, 5\}$) and outcome ($\mathcal{I}_{\text{out}} = \{1, 2, 3, 4, 5\}$) models in this simulation.

A.2 Additional Simulation Results

Figure 3 shows the distribution of worst-group MSE across the 100 simulation replications. ROME-EM reduces the mean worst-group MSE from 36.56 under pooled regression to 32.59, corresponding to an approximately 11% reduction (paired t-test: $p < 0.001$).

Figures 4, 5, and 6 present the additional simulation results as complementary to Figure 1 in the main text.

B Real-Data Details

Law School Admissions Council (Wightman, 1999) We use 'zfygpa' as Y, features 'race1_black' (S_1), 'gender_male' (S_2) and 'age' (S_3) as $\boldsymbol{S}$; 'lsat','ugpa','fam_inc', 'tier' and 'full-time' as $\boldsymbol{A}$.

Communities and Crime (Redmond, 2002) We use 'ViolentCrimesPerPop' as Y, features 'racepctblack' (S_1), 'racePctHisp' (S_2) and 'agePct12t21' (S_3) as $\boldsymbol{S}$; and features 'PctUnemployed', 'medIncome', 'PopDens', 'PolicPerPop', 'MedRent', 'PctFam2Par','PctIlleg', 'LandArea' and 'pctWWage' as $\boldsymbol{A}$.

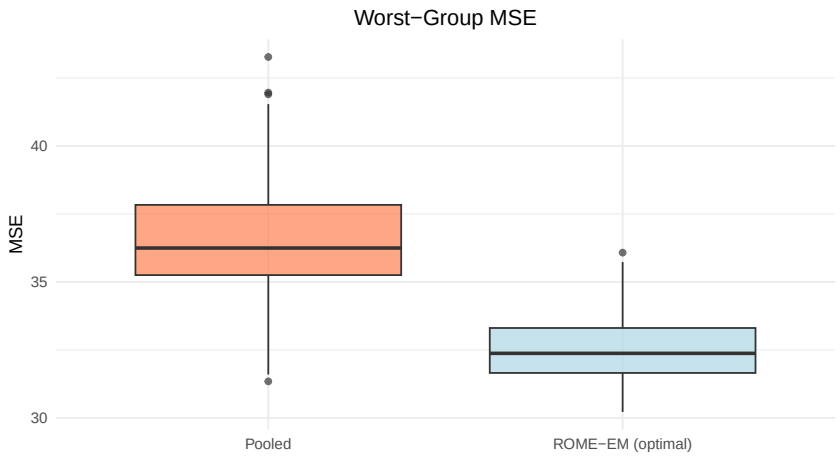

Figure 3: Box plots comparing worst-group MSE over 100 simulations.

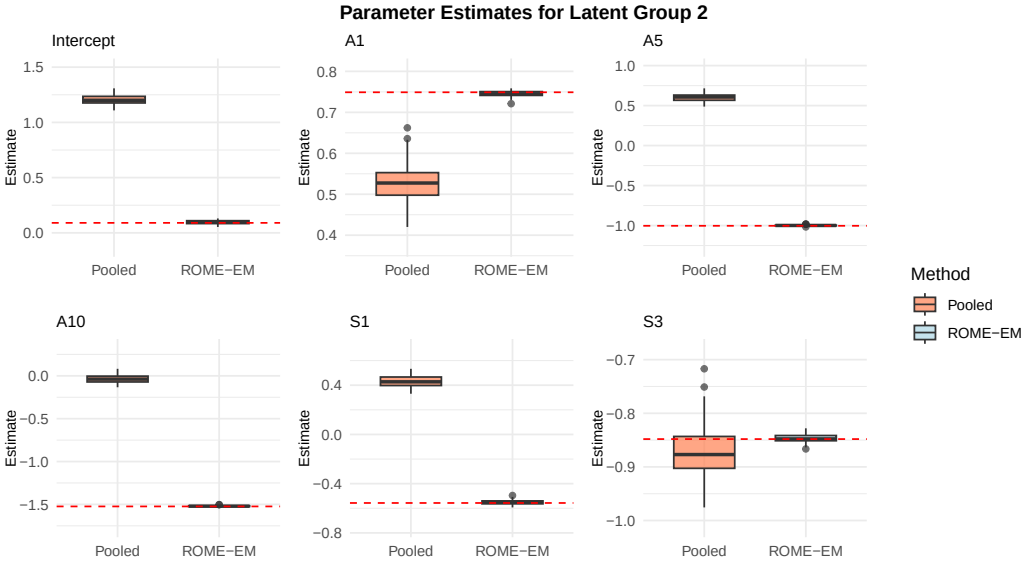

Figure 4: Box plots comparing parameter recovery by ROME-EM and pooled regression for latent group 2 (over 100 simulations). Red dashed lines denote ground truth values for each parameter.

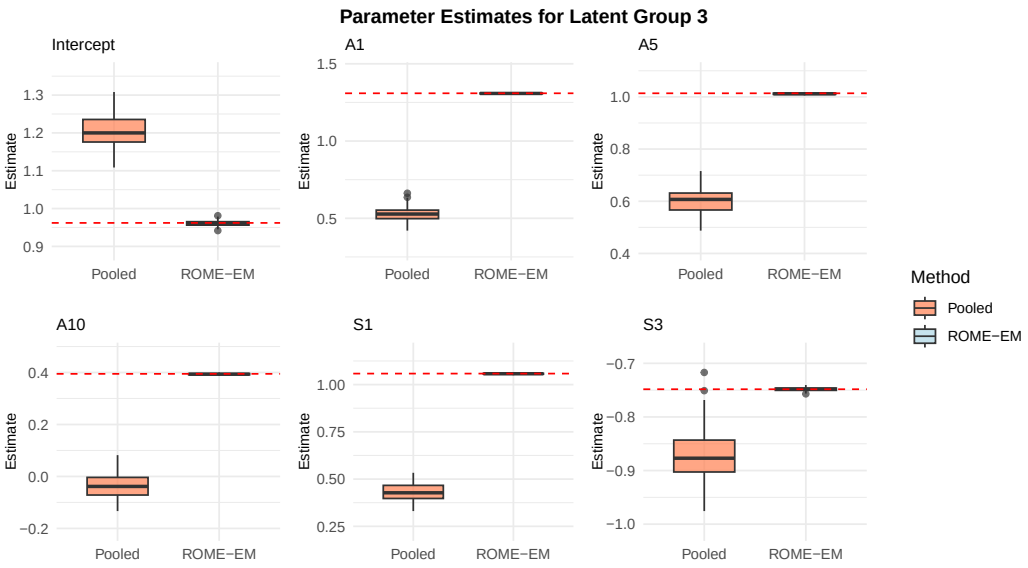

Figure 5: Box plots comparing parameter recovery by ROME-EM and pooled regression for latent group 3 (over 100 simulations). Red dashed lines denote ground truth values for each parameter.

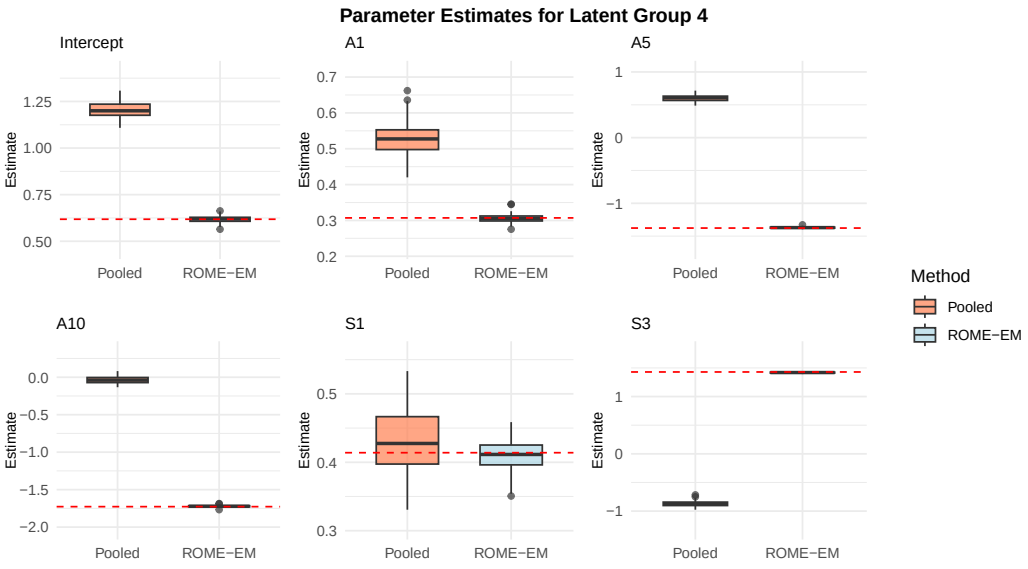

Figure 6: Box plots comparing parameter recovery by ROME-EM and pooled regression for latent group 4 (over 100 simulations). Red dashed lines denote ground truth values for each parameter.

American Community Survey Public Use Microdata Sample (Bureau, 2003) We use 'PINCP' as Y, features 'SEX' (S_1), 'RAC1P'(S_2), 'AGEP' (S_3) and 'SCHL' (S_4) as $\boldsymbol{S}$; and features 'COW', 'MAR', 'OCCP', 'POBP', 'RELP', 'WKHP' as $\boldsymbol{A}$.

C Ablation Studies

C.1 Impact of DRO Parameter

Figures 7 and 8 present the additional results for α ablation studies as complementary to Figure 2 in the main text.

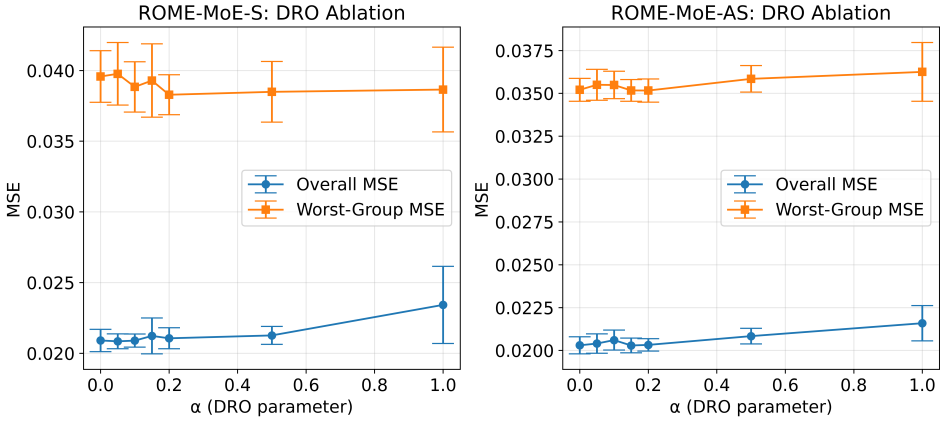

Figure 7: Effect of DRO parameter α on ROME-MoE performance (Communities & Crime dataset, 10 seeds). Both variants show worst-group improvements with minimal overall performance degradation as α increases from 0 (standard training) to 1 (pure worst-group optimization).

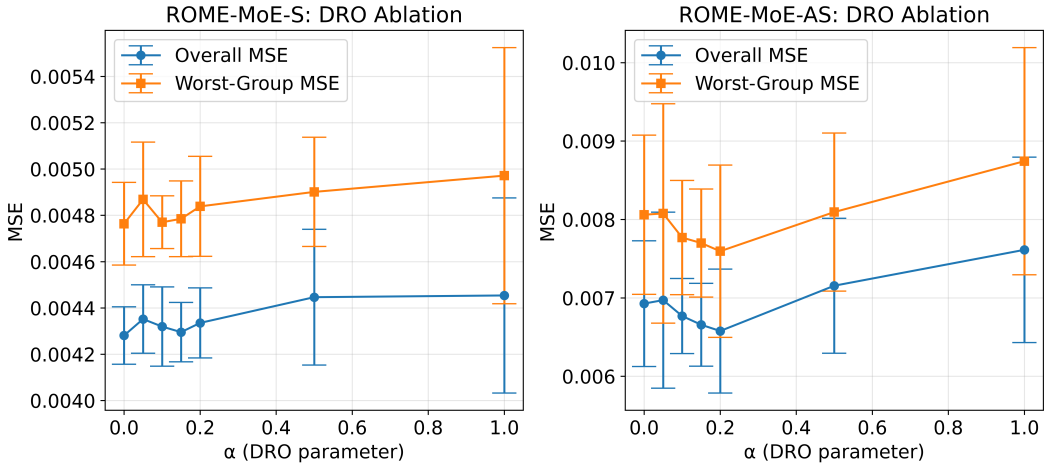

Figure 8: Effect of DRO parameter α on ROME-MoE performance (American Community Survey dataset, 10 seeds). Both variants show worst-group improvements with minimal overall performance degradation as α increases from 0 (standard training) to 1 (pure worst-group optimization).

C.2 Impact of Evaluation Subgroups

Evaluation subgroups are formed by partitioning sensitive attributes: categorical variables retain their discrete values, while continuous variables are discretized using median or quartile splits. Besides Table 1 in main text, where we use quartile cuts for (S_2, S_3, quartile) for law dataset, (S_2, S_3,

median) for crime dataset and S_2 for American Community Survey dataset, where feature details are available in B, we conduct ablation studies by further using different evaluation subgroups.

Table 2: Ablation study on evaluation subgroup schemes for the Law School dataset. Results show worst-group performance across different sensitive attribute combinations, averaged over 10 random seeds (mean ± standard error). Best-performing fair models are shown in **bold**. Statistical significance is assessed using paired two-sided t-tests across the 10 matched random seeds, comparing each ROME variant with the best-performing fair baseline for each evaluation subgroup scheme: $^{*}p < 0.05$, $^{**}p < 0.01$, and $^{***}p < 0.001$; indicates no significant difference ($p > 0.05$).

Evaluation Subgroups	Model	Fair	Worst-Group MSE	Worst-Group R^2
S_1, S_3, median	Baseline MLP		0.7640 ± 0.0007	0.1729 ± 0.0656
	Baseline MLP - Fair	✓	0.7795 ± 0.0047	0.0402 ± 0.0827
	ARL	✓	0.7744 ± 0.0034	-0.0202 ± 0.0102
	FairReg-BGL	✓	0.7730 ± 0.0024	-0.0227 ± 0.0098
	Vanilla MoE		0.7656 ± 0.0010	0.1719 ± 0.0663
	ROME-MoE-S	✓	$0.7628 \pm 0.0010^{**}$	$\mathbf{0.1710 \pm 0.0664}^{***}$
	ROME-MoE-AS	✓	$\mathbf{0.7622 \pm 0.0015}^{**}$	$0.1581 \pm 0.0689^{**}$
S_2, S_3, median	Baseline MLP		0.7674 ± 0.0010	0.1276 ± 0.0006
	Baseline MLP - Fair	✓	0.7825 ± 0.0015	0.1029 ± 0.0011
	ARL	✓	0.7811 ± 0.0013	0.1037 ± 0.0008
	FairReg-BGL	✓	0.7787 ± 0.0016	0.1010 ± 0.0007
	Vanilla MoE		0.7681 ± 0.0015	0.1277 ± 0.0006
	ROME-MoE-S	✓	$0.7675 \pm 0.0019^{***}$	$0.1210 \pm 0.0020^{***}$
	ROME-MoE-AS	✓	$\mathbf{0.7644 \pm 0.0016}^{***}$	$\mathbf{0.1235 \pm 0.0012}^{***}$

Table 3: Ablation study on evaluation subgroup schemes for the Communities & Crime dataset. Results show worst-group performance across different sensitive attribute combinations, averaged over 10 random seeds (mean ± standard error). Best-performing fair models are shown in **bold**. Statistical significance is assessed using paired two-sided t-tests across the 10 matched random seeds, comparing each ROME variant with the best-performing fair baseline for each metric and evaluation subgroup scheme: $^{*}p < 0.05$, $^{**}p < 0.01$, and $^{***}p < 0.001$; indicates no significant difference ($p > 0.05$).

Evaluation Subgroups	Model	Fair	Worst-Group MSE	Worst-Group R^2
S_1, S_2, median	Baseline MLP		0.0402 ± 0.0005	0.2106 ± 0.0081
	Baseline MLP - Fair	✓	0.0418 ± 0.0003	-0.0542 ± 0.0425
	ARL	✓	0.0381 ± 0.0003	-0.0085 ± 0.0243
	FairReg-BGL	✓	0.0379 ± 0.0002	0.0032 ± 0.0104
	Vanilla MoE		0.0404 ± 0.0007	0.2100 ± 0.0084
	ROME-MoE-S	✓	0.0407 ± 0.0005	$\mathbf{0.2005 \pm 0.0061}^{***}$
	ROME-MoE-AS	✓	$\mathbf{0.0369 \pm 0.0002}^{*}$	$0.1406 \pm 0.0108^{***}$
S_1, S_3, median	Baseline MLP		0.0332 ± 0.0004	0.1847 ± 0.0076
	Baseline MLP - Fair	✓	0.0385 ± 0.0004	0.1352 ± 0.0156
	ARL	✓	0.0362 ± 0.0003	0.1322 ± 0.0082
	FairReg-BGL	✓	0.0360 ± 0.0002	0.1405 ± 0.0028
	Vanilla MoE		0.0339 ± 0.0007	0.1885 ± 0.0079
	ROME-MoE-S	✓	$0.0339 \pm 0.0004^{***}$	$\mathbf{0.1796 \pm 0.0051}^{***}$
	ROME-MoE-AS	✓	$\mathbf{0.0319 \pm 0.0002}^{***}$	0.1520 ± 0.0050

Table 4: Ablation study on evaluation subgroup schemes for the American Community Survey Public Use Microdata Sample dataset. Results show worst-group performance across different sensitive attribute combinations, averaged over 10 random seeds (mean ± standard error). Best-performing fair models are shown in **bold**. Statistical significance is assessed using paired two-sided t-tests across the 10 matched random seeds, comparing each ROME variant with the best-performing fair baseline for each metric and evaluation subgroup scheme: $^{*}p < 0.05$, $^{**}p < 0.01$, and $^{***}p < 0.001$; indicates no significant difference ($p > 0.05$).

Evaluation Subgroups	Model	Fair	Worst-Group MSE	Worst-Group R^2
S_1	Baseline MLP		0.0047 ± 0.0000	0.4870 ± 0.0037
	Baseline MLP - Fair	✓	0.0055 ± 0.0001	0.4026 ± 0.0054
	ARL	✓	0.0056 ± 0.0000	0.4008 ± 0.0040
	FairReg-BGL	✓	0.0056 ± 0.0001	0.3959 ± 0.0071
	Vanilla MoE		0.0052 ± 0.0000	0.4451 ± 0.0038
	ROME-MoE-S	✓	$\mathbf{0.0049 \pm 0.0001}^{***}$	$\mathbf{0.4678 \pm 0.0060}^{***}$
	ROME-MoE-AS	✓	$0.0051 \pm 0.0001^{***}$	$0.4499 \pm 0.0065^{***}$
S_1 and S_2	Baseline MLP		0.0049 ± 0.0000	0.4445 ± 0.0053
	Baseline MLP - Fair	✓	0.0056 ± 0.0001	0.3764 ± 0.0067
	ARL	✓	0.0056 ± 0.0000	0.3827 ± 0.0067
	FairReg-BGL	✓	0.0057 ± 0.0001	0.3822 ± 0.0067
	Vanilla MoE		0.0052 ± 0.0000	0.4039 ± 0.0049
	ROME-MoE-S	✓	$\mathbf{0.0050 \pm 0.0001}^{***}$	$\mathbf{0.4328 \pm 0.0080}^{**}$
	ROME-MoE-AS	✓	$0.0052 \pm 0.0001^{***}$	0.4074 ± 0.0080

D Neural Network Details

D.1 Neural Network Architecture

All MoE-based models (Vanilla MoE, ROME-MoE-S, ROME-MoE-AS) use standard two-layer multilayer perceptrons (MLPs) for both the experts and the gating network. Each expert has the structure `Linear{ReLU{Linear` and outputs a scalar prediction. The Vanilla MoE experts take (A, S) as input, while the fair variants (ROME-MoE-S / ROME-MoE-AS) use only A as expert input. The gating network is also a two-layer MLP, taking either S (ROME-MoE-S) or (A, S) (ROME-MoE-AS) as input and producing G logits, followed by a softmax to obtain mixture weights. The final prediction is the weighted combination of expert outputs. Hyperparameter search spaces and selected values (e.g., hidden sizes $\{32, 64, 128\}$, number of experts $G \in \{2, 3, 4\}$) are provided in Tables 5, 6 and 7.

All baseline MLP models use a standard two-layer multilayer perceptron (MLP) of the form Linear → ReLU → Linear, producing a scalar prediction. The variants differ only in their input features and training objective: Baseline MLP–Full (unfair): experts receive the full feature vector (A, S); Baseline MLP–Fair: experts receive only A, matching the fairness constraint used in ROME-MoE. Hyperparameter search spaces (hidden sizes {32, 64, 128}, learning rates, etc.) are identical to those used for MoE and are summarized in Tables 5–7.

Fair Regression-BGL. We implement the bounded group loss (BGL) formulation of Fair Regression (Agarwal et al., 2019) using the `BoundedGroupLoss` and `ExponentiatedGradient` reductions in Fairlearn. BGL minimizes overall prediction loss subject to an upper bound ζ on the loss within each predefined sensitive group. We use squared loss and an $\boldsymbol{A}$-only Linear–ReLU–Linear MLP as the base regression learner, with hidden sizes 32, 64, and 32 for Law School, Communities & Crime, and ACS PUMS, respectively. We fix the learning rate at 10^{-3}, use a batch size of 64, and train for 30 epochs. Sensitive attributes are used to construct the BGL constraints but are not provided as inputs to the regression learner.

For each dataset, BGL is trained using the sensitive-group definition corresponding to the main evaluation setting: (S_2, S_3) with quartile discretization for Law School, (S_2, S_3) with median discretization for Communities & Crime, and S_2 for ACS PUMS. We first perform a coarse search over

the BGL upper bound ζ and then refine the search around the feasible region using the held-out validation split. Among approximately feasible configurations, we select ζ according to worst-group validation MSE, using overall validation MSE as a tie-breaker. The final refined search spaces and selected values are reported in Tables 5, 6, and 7. For evaluation, we use the deterministic weighted prediction from the mixture of predictors returned by the exponentiated-gradient reduction.

D.2 Hyperparameter Tuning

Hyperparameters were selected using a held-out 20% subset of the original training data, with the test set held out throughout model selection. The search spaces and final selected values for each dataset are reported in Tables 5, 6, and 7. For ARL, we tune the learner and adversary learning rates while fixing the learner architecture. For Fair Regression-BGL, we fix the base learner and tune the BGL upper bound ζ. After hyperparameter selection, models are refit using the full training set and evaluated on the held-out test set over 10 random seeds.

Table 5: Hyperparameter tuning results for Law School Admissions Council

Hyperparameter	**Search Space**	**Best Value**
BaselineMLP		
Learning Rate	{1e-2, 1e-3, 1e-4}	1e-4
Hidden Size	{32, 64, 128}	64
BaselineMLP-Fair		
Learning Rate	{1e-2, 1e-3, 1e-4}	1e-3
Hidden Size	{32, 64, 128}	32
ARL		
Learner Learning Rate	{1e-3, 1e-4}	1e-3
Adversary Learning Rate	{1e-3, 3e-4, 1e-4, 3e-5}	1e-4
Hidden Size	32	32
Fair Regression-BGL		
BGL Upper Bound ζ	{0.72, 0.74, 0.76, 0.78, 0.80}	0.78
Vanilla MoE		
Learning Rate	{1e-2, 1e-3, 1e-4}	1e-4
Hidden Size (expert)	{32, 64, 128}	32
Hidden Size (gating)	{32, 64, 128}	64
Number of Experts	{2, 3, 4}	4
ROME-MoE-S		
Learning Rate	{1e-2, 1e-3, 1e-4}	1e-4
Hidden Size (expert)	{32, 64, 128}	128
Hidden Size (gating)	{32, 64, 128}	16
Number of Experts	{2, 3, 4}	2
ROME-MoE-AS		
Learning Rate	{1e-2, 1e-3, 1e-4}	1e-4
Hidden Size (expert)	{32, 64, 128}	64
Hidden Size (gating)	{32, 64, 128}	64
Number of Experts	{2, 3, 4}	2

Table 6: Hyperparameter tuning results for Communities and Crime dataset

Hyperparameter	**Search Space**	**Best Value**
BaselineMLP		
Learning Rate	{1e-2, 1e-3, 1e-4}	1e-2
Hidden Size	{32, 64, 128}	128
BaselineMLP-Fair		
Learning Rate	{1e-2, 1e-3, 1e-4}	1e-2
Hidden Size	{32, 64, 128}	64
ARL		
Learner Learning Rate	{1e-3, 1e-4}	1e-3
Adversary Learning Rate	{1e-3, 3e-4, 1e-4, 3e-5}	3e-4
Hidden Size	64	64
Fair Regression-BGL		
BGL Upper Bound ζ	{0.0260, 0.0265, 0.0270, 0.0275, 0.0280}	0.0270
Vanilla MoE		
Learning Rate	{1e-2, 1e-3, 1e-4}	1e-2
Hidden Size (expert)	{32, 64, 128}	64
Hidden Size (gating)	{32, 64, 128}	32
Number of Experts	{2, 3, 4}	2
ROME-MoE-S		
Learning Rate	{1e-2, 1e-3, 1e-4}	1e-2
Hidden Size (expert)	{32, 64, 128}	64
Hidden Size (gating)	{32, 64, 128}	16
Number of Experts	{2, 3, 4}	3
ROME-MoE-AS		
Learning Rate	{1e-2, 1e-3, 1e-4}	1e-3
Hidden Size (expert)	{32, 64, 128}	32
Hidden Size (gating)	{32, 64, 128}	16
Number of Experts	{2, 3, 4}	2

Table 7: Hyperparameter tuning results for American Community Survey Public Use Microdata Sample dataset

Hyperparameter	**Search Space**	**Best Value**
BaselineMLP		
Learning Rate	{1e-2, 1e-3, 1e-4}	1e-3
Hidden Size	{32, 64, 128}	64
BaselineMLP-Fair		
Learning Rate	{1e-2, 1e-3, 1e-4}	1e-3
Hidden Size	{32, 64, 128}	32
ARL		
Learner Learning Rate	{1e-3, 1e-4}	1e-3
Adversary Learning Rate	{1e-3, 3e-4, 1e-4, 3e-5}	1e-4
Hidden Size	32	32
Fair Regression-BGL		
BGL Upper Bound ζ	{0.00520, 0.00530, 0.00540, 0.00545, 0.00550}	0.00550
Vanilla MoE		
Learning Rate	{1e-2, 1e-3, 1e-4}	1e-4
Hidden Size (expert)	{32, 64, 128}	128
Hidden Size (gating)	{32, 64, 128}	16
Number of Experts	{2, 3, 4}	4
ROME-MoE-S		
Learning Rate	{1e-2, 1e-3, 1e-4}	1e-2
Hidden Size (expert)	{32, 64, 128}	128
Hidden Size (gating)	{32, 64, 128}	32
Number of Experts	{2, 3, 4}	2
ROME-MoE-AS		
Learning Rate	{1e-2, 1e-3, 1e-4}	1e-4
Hidden Size (expert)	{32, 64, 128}	128
Hidden Size (gating)	{32, 64, 128}	32
Number of Experts	{2, 3, 4}	4