DIRECTLY OPTIMIZING MEAN DEMOGRAPHIC PARITY FOR NONLINEAR REGRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

We focus on regression settings where the fairness goal is to equalize average predictions across values of a sensitive attribute, a criterion known as mean demographic parity. Directly optimizing this criterion is difficult because it depends on a conditional mean that is unknown and changes during training. Common dependence penalties and adversarial methods do not estimate this conditional mean; instead, they push predictions toward full independence. This stronger constraint can reduce accuracy even when average predictions are already equal. Existing conditional-mean methods are limited to linear predictors or low-dimensional sensitive attributes. We enable direct optimization of mean demographic parity using DPVar, a fairness measure defined as the variance of the conditional mean prediction. Because the conditional mean must be estimated as the predictor changes, optimizing DPVar leads to a functional bilevel problem. We develop two solvers: FBO, which uses a closed-form hypergradient, and an iterative-differentiation (ITD) solver that differentiates through updates of the conditional-mean estimator. Unlike previous conditional-mean methods, our approach applies to nonlinear predictors and high-dimensional continuous sensitive attributes. Across a semi-synthetic benchmark built from 21 tabular regression datasets and Communities & Crime data, FBO and ITD recover competitive or better accuracy–DPVar trade-offs than existing methods.

1 INTRODUCTION

Predictive machine learning models are effective because they exploit statistical signals that reduce prediction error. Unfortunately standard training does not distinguish desirable signals from those associated with attributes deemed 'sensitive" for legal, ethical, or policy reasons (Barocas & Selbst, 2016; Mehrabi et al., 2021). Because models are often required to be 'fair" with respect to these attributes, one must choose a fairness criterion and optimize its trade-off with predictive quality. Fairness encompasses a *family* of criteria (Mehrabi et al., 2021): demographic parity (DP) compares predictions across sensitive groups (Jiang et al., 2022); equalized odds and equal opportunity condition on the true outcome (Hardt et al., 2016); and individual and counterfactual fairness require consistency across similar individuals or hypothetical interventions (Kusner et al., 2017; Barocas et al., 2023). These criteria are neither equivalent nor always jointly attainable (Chouldechova, 2017; Kleinberg et al., 2017). We focus on the demographic parity family, motivated by settings where the goal is to control systematic differences in predictions across sensitive groups.

A typical problem considered in the DP literature aims at learning a map f_ω that predicts a target Y from non-sensitive features X from supervised data, while enforcing a DP condition w.r.t. to sensitive attribute A. In its *strong* form, DP requires statistical independence between the predictor $f_\omega(X)$ and sensitive attribute A(Kamishima et al., 2012; Gretton et al., 2008; Zhang et al., 2018; Edwards & Storkey, 2016), thereby equalizing the entire conditional distribution of predictions across values of A. If the fairness concern is a systematic shift in *average* prediction, strong DP also constrains spread, tails, and shape, potentially degrading predictive performance. In credit scoring and job recommendation (Vladimirova et al., 2024), for example, one may seek comparable average credit scores or salary recommendations across demographic profiles without requiring identical distributions (see Section D). *Mean demographic parity* relaxes strong DP by requiring only that $\mathbb{E}[f_\omega(X) \mid A]$ be approximately constant (Wei et al., 2023; Jiang et al., 2022; Yazdani-Jahromi et al., 2024). This condition can be naturally encouraged by a penalty term controlling the variance of the

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

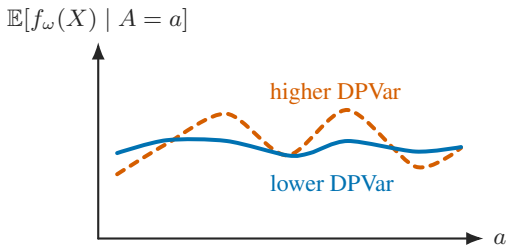


Figure 1: Conceptual illustration of DPVar. A predictor with higher DPVar has a conditional-mean profile $a \mapsto \mathbb{E}[f_\omega(X) \mid A = a]$ that varies more strongly with the sensitive attribute. Lower DPVar corresponds to a flatter conditional-mean profile.

conditional mean across A, which we call *DPVar* and that was previously considered in the literature in restricted settings (Komiyama et al., 2018; Wei et al., 2023). As Figure 1 illustrates, DPVar measures how average predictions vary with A and equals zero exactly when mean demographic parity holds.

DPVar and related fairness measures depend on the conditional mean of predictions, which is generally unavailable in closed form and must be estimated. For discrete A, groupwise sample means suffice (Charpentier et al., 2023; Glasserman & Li, 2024; Wei et al., 2023); for continuous, multidimensional A, estimation is more challenging. Komiyama et al. (2018) address continuous sensitive attributes through a coefficient-of-determination (CoD) constraint and derive tractable algorithms for linear and kernelized least-squares predictors. However, their formulation does not directly accommodate neural networks, which can model more complex relationships. Jiang et al. (2022) instead penalize the average absolute deviation of the conditional mean from the overall mean, estimating it through kernel smoothing and numerical integration. This accommodates continuous sensitive attributes and general predictors, including neural networks. In higher dimensions, however, kernel estimation does not scale well to high dimensions, which limits the applicability of such approaches to small dimensional sensitive attributes A (only up to 2 dimensions were considered in practice).

In this work, we address the above limitations by showing that direct optimization of DPVar has a natural functional-bilevel (Petrulionyte et al., 2024) structure: the inner problem estimates the predictor-dependent conditional mean, while the outer problem optimizes prediction accuracy and mean parity jointly. The inner problem estimates the conditional mean $h^\star_\omega(a) = \mathbb{E}[f_\omega(X) \mid A = a]$ by regressing predictions $f_\omega(X)$ on A in function space. Such estimate is provided to the outer problem which minimizes the predictive loss penalized by DPVar, while accounting for the dependence of $h^\star_\omega$ on the predictor parameters ω. This approach accommodates continuous, multidimensional sensitive attributes and permits neural networks to represent both the predictor and its conditional mean. It also avoids kernel smoothing, whose estimation accuracy can deteriorate as the dimension of A grows. We develop scalable first-order algorithms for the resulting bilevel problem and evaluate them against state-of-the-art baselines on datasets of increasing difficulty. Specifically, our contributions are as follows:

1. We turn direct DPVar minimization into a tractable learning problem for nonlinear predictors and continuous, high-dimensional sensitive attributes by estimating the required conditional mean as an inner regression function.
2. For squared-loss inner regression, we derive an explicit population hypergradient whose functional Hessian reduces to $2I$, yielding a scalable plug-in FBO update without parameter-space Hessian inversion; we additionally evaluate ITD as a complementary unrolled solver.
3. We introduce a semi-synthetic benchmark dataset of 21 public regression tasks, selected to exhibit a nontrivial sensitive-attribute component of the prediction.
4. We demonstrate competitive performance against state-of-the-art baselines, on synthetic tasks, the semi-synthetic benchmark and a real dataset from a Communities & Crime study using observed ten-dimensional continuous demographic attribute.

The remainder of the paper proceeds as follows. Section 2 reviews related work. Section 3 introduces the functional bilevel problem and the proposed algorithms. Section 4 reports experiments.

2 Related Work

Demographic parity and mean parity. Most demographic-parity methods seek to make predictions independent of the sensitive attribute. This requirement has been approached through constraints and reductions (Zafar et al., 2017; Agarwal et al., 2018), dependence penalties based on kernels or mutual information (Pérez-Suay et al., 2017; Li et al., 2022; Kamishima et al., 2012), and adversarial training (Edwards & Storkey, 2016; Zhang et al., 2018). For continuous sensitive attributes, methods based on Rényi's maximal correlation criterion likewise target dependence between predictions and A (Mary et al., 2019; Grari et al., 2020). Optimal-transport methods pursue the same distributional notion of parity by mapping group-conditional prediction distributions toward a Wasserstein barycenter (Chzhen et al., 2020; Le Gouic et al., 2020; Chzhen & Schreuder, 2022). Related barycenter methods address other performance measures (Gaucher et al., 2023). Our objective concerns a weaker requirement: the conditional *mean* of a real-valued prediction should be constant across values of A. Mean parity leaves other features of the prediction distribution unconstrained. It has been studied directly in RKHS regression for discrete sensitive attributes (Wei et al., 2023); we encourage it by penalizing DPVar and propose a method that handle continuous high-dimensional attributes.

Continuous sensitive attributes and conditional-mean criteria. Work closest to our setting handles continuous sensitive attributes by exploiting the structure of the predictor, smoothing over values of A, or partitioning A into groups. Komiyama et al. (2018) constrain the proportion of prediction variance explained by multiple numeric sensitive attributes, using optimization tailored to linear or kernelized least-squares predictors, but which does not directly extend to neural networks. Jiang et al. (2022) measure the average absolute difference between the conditional prediction mean and the overall mean estimated through kernel smoothing and numerical integration over sensitive values, and thus does not scale well in high dimensions. Finally, Shilova et al. (2025) identify partitions of continuous sensitive attributes that reveal disparities between the resulting groups. Their partitioning algorithms are developed for one- and two-dimensional attributes, while scaling the search to higher dimensions remains an open challenge. In this work, we measure conditional-mean variation with respect to the original, ungrouped A and estimate that mean through a learned regression function, allowing neural predictors without discretizing A or applying kernel smoothing over it.

Bilevel optimization for fairness. Bilevel optimization has been used in fair learning to adjust training weights, batch composition, and other choices in the learning procedure (Roh et al., 2021; Ozdayi et al., 2022; Yazdani-Jahromi et al., 2024; Tanji et al., 2026). For DPVar, the nested structure arises instead because evaluating the fairness criterion requires a conditional mean, which can itself be defined as the solution of a regression problem. This connects our setting to functional bilevel optimization, where the inner solution is a function (Petrulionyte et al., 2024).

Figure 2 summarizes the relationships among these lines of work; further detailed discussions of the various methods appear in Section B.

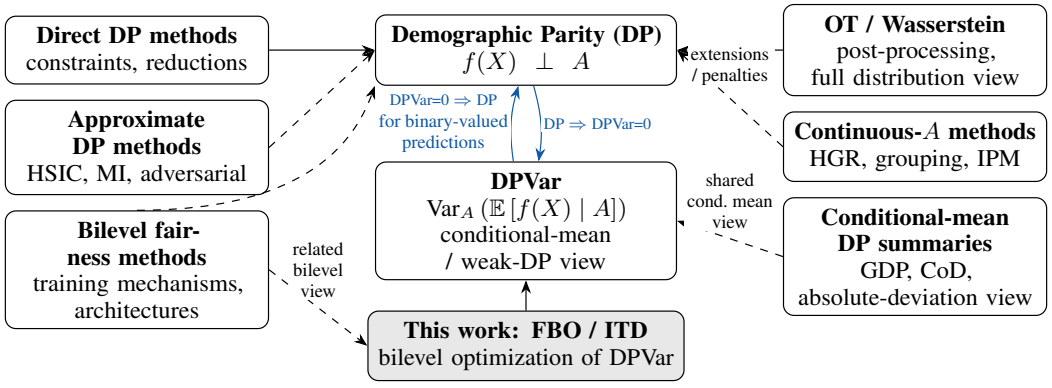


Figure 2: Positioning of DPVar relative to demographic parity (DP) and existing fairness methods. The blue arrows show the relation between full distributional DP and conditional-mean DP. Solid method arrows indicate direct optimization of the target object; dashed arrows indicate related relaxations, penalties, or approximate formulations.

3 From DPVar to a Practical Functional Bilevel Algorithm

3.1 Problem formulation

Setup and objective. We consider the learning task of predicting a target variable $Y \in \mathbb{R}$ from non-sensitive features $X \in \mathbb{R}^{k_X}$, while encouraging mean demographic parity w.r.t. a sensitive attribute $A \in \mathbb{R}^{k_A}$. Predictions are made using a model $f_\omega : \mathbb{R}^{k_X} \to \mathbb{R}$, with parameter vector $\omega \in \Omega$ belonging to some Euclidean subset Ω and to be learned from data. Here, data comes in the form of triplets (X, Y, A) drawn from a probability distribution P, with P_A denoting the marginal law of A. Our learning problem, in population, consists in minimizing an objective $\omega \mapsto F(\omega)$ that balances mean squared prediction error against our proposed criterion, DPVar, for measuring violations of mean demographic parity by the variation of the conditional mean prediction across A

$$F(\omega) = \mathbb{E}_P\left[(f_\omega(X) - Y)^2\right] + \alpha \ \mathrm{DPVar}(\omega), \qquad \mathrm{DPVar}(\omega) = \mathrm{Var}_{A\sim P_A}\left(\mathbb{E}_P[f_\omega(X) \mid A]\right), \quad (1)$$

where α is a non-negative hyper-parameter that sets the accuracy–mean-parity trade-off. The difficulty with the above problem is that DPVar requires the unknown function $a \mapsto \mathbb{E}_P[f_\omega(X) \mid A = a]$, which also changes as ω changes.

Functional bilevel formulation. We propose to address this difficulty by estimating the conditional mean through a regression of the current predictions on A. this allows a flexible model to learn shared structure across values of A, potentially resulting in efficient estimation. To distinguish the two roles of the data in training, we let `IN` and `OUT` be independent samples from the population law P. We then write P_{in} and P_{out} for their population laws and keep P_A for their common marginal law of A. The inner sample fits the conditional mean; the outer sample evaluates the prediction loss and the fairness penalty. At the population level, the function being fitted is

$$h^\star_\omega(a) = \mathbb{E}_{(X,A)\sim P_{\mathrm{in}}}[f_\omega(X) \mid A = a]. \quad (2)$$

Assuming $f_\omega(X)$ is square-integrable, and setting $\mathcal{H} = L^2(P_A)$, the least-squares projection property of conditional expectation gives the following equivalent characterization, with uniqueness understood up to P_A-almost-sure equality:

$$h^\star_\omega \in \arg\min_{h\in\mathcal{H}} \mathbb{E}_{(X,A)\sim P_{\mathrm{in}}}\left[(f_\omega(X) - h(A))^2\right]. \quad (3)$$

This characterization turns estimation of a conditional expectation into ordinary regression on the `IN` sample. Substituting the regression solution into the penalized objective thus yields:

$$\min_\omega \ \mathbb{E}_{P_{\mathrm{out}}}[(f_\omega(X) - Y)^2] + \alpha \, \mathrm{Var}_{A\sim P_A}(h^\star_\omega(A)).$$

This above problem is a bilevel optimization problem, since the objective depends on $h^\star_\omega$, the solution of the inner regression problem in Equation (3), which, in turn, depends implicitly on the parameter ω. It is *functional* because the population inner variable belongs to the function space $\mathcal{H}$, even though, in practice, we will approximate $h^\star_\omega$ with a neural network with finite-dimensional parameter during training (Petrulionyte et al., 2024).

3.2 Gradient-based algorithms

As we will consider predictors that are neural networks, we are interested in optimizing their parameters ω using gradient methods. To this end, one must estimate $\nabla_\omega F(\omega)$, often called the *hypergradient* in the context of bilevel optimization. It must account for the response of $h^\star_\omega$ to a change in ω. We consider two ways to obtain a usable gradient, both of which require estimating $h^\star_\omega$ using an additional model of parameters ϕ and which, by abuse of notation, will be denoted as h_ϕ. *Functional bilevel optimization* (FBO) derives a population hypergradient in the function space $\mathcal{H}$ and substitutes the estimate h_ϕ to $h^\star_\omega$. *Iterative differentiation* (ITD) differentiates through a finite sequence of updates to the parameters ϕ of the model h_ϕ.

Functional bilevel optimization (FBO). When fitting the model h_ϕ using the regression objective in Equation (3), the optimal ϕ depends implicitly on the outer parameter ω. Thus one might consider using methods such as approximate implicit-differentiation (AID) which would require a inverting the Hessian of the inner-loss wrt the ϕ. When using an overparameterized neural network, such a Hessian need not be invertible or well conditioned, so that route can be sensitive to approximation and regularization choices. The functional formulation does not require differentiating w.r.t. ϕ to

estimate the sensitivity of the optimal regressor $h^\star_\omega$ to ω (Petrulionyte et al., 2024). Instead, it needs to compute the Hessian of the inner objective w.r.t. to the function h in the function space $\mathcal{H}$. For the squared loss considered in Equation (3), the Hessian with respect to the *function* $h \in L^2(P_A)$ is $2I$, which allows deriving the following simple expression for the hyper-gradient.

Proposition 3.1 (Population gradient for squared-loss DPVar)**.** *Suppose $f_\omega(X)$, Y, and the relevant derivatives are square-integrable; and differentiation can be interchanged with the expectations and conditional expectation below. Let $h^\star_\omega$ solve Equation (3) and define $\mu_\omega = \mathbb{E}_{A\sim P_A}[h^\star_\omega(A)]$. Then the gradient of Equation (1) can be written as*

$$\nabla_\omega F(\omega) = 2\,\mathbb{E}_{\text{out}}[(f_\omega(X) - Y)\,\partial_\omega f_\omega(X)] + 2\alpha\,\mathbb{E}_{\text{in}}[(h^\star_\omega(A) - \mu_\omega)\,\partial_\omega f_\omega(X)]\,.$$

The proof is in Appendix C. The second expectation contains the dependence of the optimal conditional mean on ω, even though it does not explicitly differentiate the fitted regressor h_ϕ. FBO substitutes h_ϕ for $h^\star_\omega$, estimates μ_ω from `OUT`, and averages the fairness term over `IN`; it also estimates the accuracy term on `OUT`. During this plug-in update, h_ϕ and the estimated mean are held fixed, as prescribed by the derived formula. The resulting object is a gradient estimate that can become more accurate by increasing the quality of the fit h_ϕ and sample size.

Iterative differentiation (ITD). Instead of using the above closed-form identity for the hypergradient, ITD starts from the current ϕ, takes K additional gradient steps on the empirical inner regression loss, evaluates the accuracy and DPVar terms on `OUT` using the resulting regressor, and differentiates that outer objective through those K steps (Pedregosa, 2016). This follows how the fitted conditional mean changes with ω over the unrolled computation. The finite K introduces truncation error and increases time and memory costs, but the procedure still showed good performance, in practice.

Training loop. Both methods first fit h_ϕ for M steps on `IN`, then update ω using the chosen hypergradient estimate: ITD additionally unrolls K inner steps for its outer update; FBO instead uses the plug-in identity in Proposition 3.1. Algorithm 1 shows the shared loop. The inner fitting and two hypergradient computations appear in Algorithms 2 and 3 in the Appendix.

Algorithm 1 Bilevel Training for DPVar

Input: initial predictor parameters ω_0; initial inner parameters ϕ_0; fairness weight α; outer stepsize η_{out}; inner stepsize η_{in}; number of outer steps N; number of inner-fit steps M; hypergradient mode HG $\in \{\text{FBO}, \text{ITD}\}$; unrolling length K
$\omega \leftarrow \omega_0$, $\phi \leftarrow \phi_0$
for $n = 0, \ldots, N-1$ **do**
  if $\alpha > 0$ **then**
    $\phi \leftarrow$ EstimateConditionalMean$(\omega, \phi, \eta_{\text{in}}, M,$ `IN`$)$
  end if
  $g \leftarrow$ ComputeHypergradient$(\omega, \phi, \alpha,$ HG$, K, \eta_{\text{in}},$ `IN`$,$ `OUT`$)$
  $\omega \leftarrow \omega - \eta_{\text{out}} g$
end for
Return: ω

4 Experiments

We evaluate methods on the trade-off between (i) predictive mean-squared error $\mathbb{E}[(f_\omega(X) - Y)^2]$ and (ii) the demographic-parity variance DPVar. Unless stated otherwise, predictors and diagnostic regressors are two-hidden-layer $\tanh$ MLPs (width 64).

Training protocol and evaluation. All methods use the same data partitioning and model-selection protocol. Training algorithms use only the `IN` and `OUT` splits; we select hyperparameters, including the fairness weight and method-specific optimization parameters, on a separate `VAL` split; and we report final results on `TEST`. To evaluate demographic-parity style deviations for continuous A in a way that is comparable across methods, we use the same diagnostic pipeline throughout: for every trained predictor f_ω, we compute DPVar by cross-fitting $h_\phi(A) \approx \mathbb{E}[f_\omega(X) \mid A]$ within the evaluation split (`VAL` for hyperparameter tuning, `TEST` for final reporting), fitting on one half and evaluating on the other and averaging across the two roles unless stated otherwise. We use α for the

population-level fairness weight in the derivation; in experiments, we distinguish the training penalty weight γ, used to train a model, from the validation weight ζ, used only for model selection along the retained validation trade-off set. Because FBO and ITD must estimate the predictor-dependent conditional mean as an inner problem, `IN` is reserved for fitting h_ϕ, while only `OUT` labels enter the predictor's accuracy loss. The single-level baselines require no such inner fit and therefore optimize prediction accuracy on pooled `IN`+`OUT` data. Thus, FBO and ITD operate with fewer labeled examples for accuracy optimization, placing them at a deliberate disadvantage in the predictive comparison.

Baselines. We compare our two bilevel methods, FBO and ITD, to standard single-level alternatives trained on `IN`+`OUT` data:

- a penalty based on the CoD statistic of Komiyama et al. (2018), which penalizes linear predictability of $f_\omega(X)$ from A[1];
- GDP*, a high-dimensional adaptation of the KDE-based Generalized Demographic Parity method of Jiang et al. (2022);
- an adversarial-debiasing baseline in which an adversary predicts A from $f_\omega(X)$ (Edwards & Storkey, 2016; Zhang et al., 2018);
- an HSIC penalty between predictions and A that follows the released closed-form method of Pérez-Suay et al. (2017).

GDP requires special treatment in our setting because its KDE estimator does not scale to high-dimensional sensitive attributes. GDP estimates the mean prediction at each sensitive-attribute value and measures how far these local means are, on average, from the overall mean prediction. Its KDE estimator approximates this integral on a Cartesian grid, whose size grows exponentially with the number of sensitive dimensions. A direct extension of the released estimator[2], using its 18 grid values per coordinate, would require $18^{10} = 3.57 \times 10^{12}$ evaluation points for our ten-dimensional Communities & Crime attribute and approximately 28.6PB merely to materialize the coordinate tensor for a batch of 200. We therefore report GDP*, which projects A onto two training-fitted principal components before applying the two-dimensional GDP estimator in both experiments.

Computational setup. All experiments were run on a heterogeneous GPU cluster with NVIDIA GPUs ranging from older GTX/Titan/P100-class cards to RTX A5000/A6000 and RTX 6000 Ada cards. Running the full method and hyperparameter-selection protocol for one tabular dataset takes roughly 2 hours. Within-run timing comparisons for Communities & Crime are reported in Section G.

4.1 Controlled experiment with known DPVar

On real data, the true conditional mean is unknown, so it is difficult to tell whether a penalty reduces the intended mean disparity or some other dependence on A. We therefore use a controlled problem whose conditional mean and DPVar can be computed from the simulator. One input feature changes in conditional mean with A, while the other changes only in conditional variance; mean parity should penalize the first, not the second. We use 2,000 samples in each of `IN`, `OUT`, `VAL`, and `TEST` across ten seeds. Every method receives exactly 17 predeclared penalty values, including zero. Iterative methods use 500 predictor updates per fit; HSIC uses its closed-form solve. All fitted models are evaluated once on `TEST`. Test DPVar is estimated directly from the simulator by Monte Carlo. All details are given in Section E.

Results. Figure 3 separates three behaviors. FBO follows the population reference over almost the full DPVar range: it learns the nonlinear conditional mean while retaining the variance-only feature for prediction. ITD moves in the same direction but covers a shorter range under the same update budget. HSIC and R^2 cannot see the constructed mean signal through linear association; GDP* and adversarial debiasing likewise do not trace the targeted frontier. This controlled experiment checks the underlying theory, while the following benchmarks provide the broader evidence.

[1] CoD: https://github.com/jkomiyama/fairregresion.
[2] GDP: https://github.com/zhimengj0326/GDP.

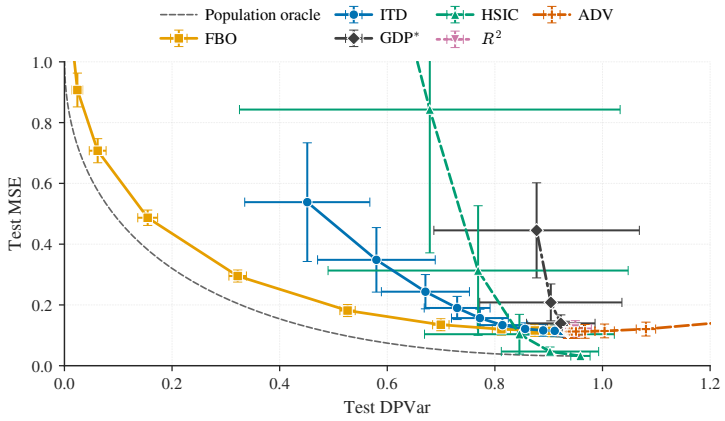


Figure 3: **Controlled accuracy–DPVar trade-off.** Points are means over ten seeds for 17 fixed penalty values per method; horizontal whiskers are 95% CI for mean test DPVar; vertical whiskers are 95% CI for mean test MSE. The dashed curve is the optimal population trade-off within the linear predictor class. Lower-left is better.

4.2 Semisynthetic benchmark on tabular datasets

We define a semi-synthetic tabular benchmark from an explicit list of regression tasks obtained from *OpenML* and *UCI* (Dua & Graff, 2019; Vanschoren et al., 2014). The datasets in the benchmark are intentionally numeric-only: categorical, Boolean, date/time, and identifier-like source columns are excluded; only source-continuous coordinates can enter A. The benchmark is not meant to claim that these coordinates are legally or socially sensitive in the original tasks. It is a controlled stress test for continuous, multidimensional sensitive attributes with nontrivial proxy information in the remaining predictors.

Benchmark membership is frozen before any fairness method is trained. Fixed construction-time rules require at least 5,000 completely finite feature/target rows in the final experiment pool, and reject classification-like, temporal or extremely sparse tasks. A held-out construction-only screen then requires a useful reference predictor, non-flat reference predictions over A, and a meaningful prediction cost when the estimated A-dependent component is removed. Applying these rules to the complete 105-task roster yields the 21 benchmark datasets, all fixed before training; no dataset is retained or discarded because of a trained method's validation range or test behavior.

Sensitive attributes. We construct A using a separate subset that is never used to train or evaluate the final models. Among continuous features, we select those that are related to the target and have information reflected in other features. This makes the benchmark nontrivial: A matters for prediction, and simply removing it from X does not eliminate its influence because proxy information remains. Applying the same fixed selection procedure across many different datasets provides a consistent and robust test of the accuracy–fairness trade-off. Full details are provided in Appendix F.

Training and model selection. For each method, we train a bounded grid of method-specific configurations and select models using only `VAL`. For each predefined trade-off weight ζ, we retain the configuration minimizing

$$\mathrm{MSE}_{\mathrm{val}} + \zeta\, \mathrm{DPVar}_{\mathrm{val}}.$$

Only these validation-selected configurations are eligible for the final test comparison. This gives all methods comparable tuning budgets and prevents uninformative extreme configurations from distorting the pareto front range. Hyperparameter grids, refinement rules, training budgets, and stopping criteria are detailed in the F.

Constrained comparison. The tabular benchmark campaign uses 11 seeds (0 through 10) across 21 datasets, for 231 total runs. For each dataset, we pool the validation DPVar values. If $d_{\min}$ and $d_{\max}$ are the extremes of this validation-only pool, the *Tight*, *Mid*, and *Loose* budgets are respectively $d_{\min} + .25(d_{\max} - d_{\min})$, $d_{\min} + .50(d_{\max} - d_{\min})$, and $d_{\min} + .75(d_{\max} - d_{\min})$. For each method, seed, and budget, we freeze the retained configuration with the lowest validation MSE among

those satisfying the validation DPVar budget. Test metrics do not enter the budget, feasibility, or configuration choices and are used only after all three are frozen.

For each dataset–seed pair, winner credit is assigned to the validation-feasible method with the lowest frozen test MSE. If no method reaches a budget for a dataset–seed pair, that pair is reported as unattainable and omitted from that seed's win-rate denominator. The figure uses the mean of the 11 seed-wise win rates as the bar height, overlays all 11 seed values as individual points, and uses whiskers for their observed minimum–maximum range.

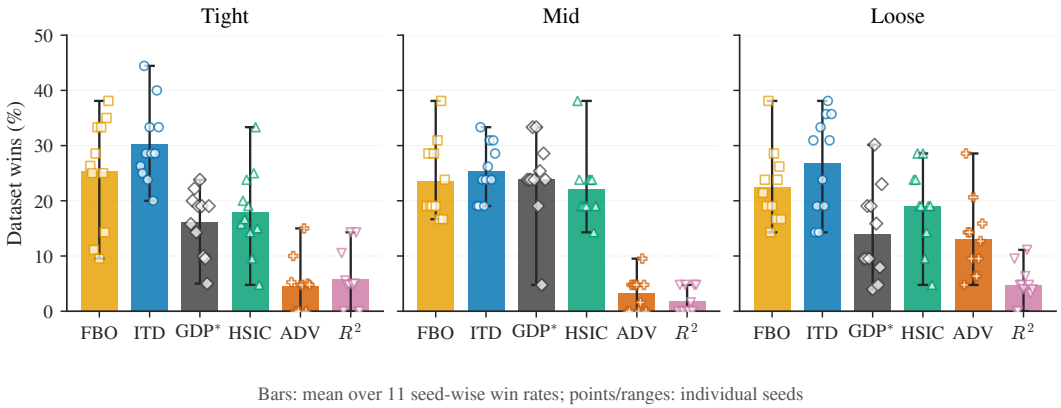


Figure 4: **Win rates on the tabular benchmark.** Tight, Mid, and Loose denote increasingly permissive validation-DPVar thresholds; Tight requires greater mean parity, while Loose allows more disparity. A win is awarded to the feasible method with the lowest test MSE, so higher is better. Bars are means over the 11 seed-wise, equal-dataset-weight win rates; points show seeds 0 through 10; whiskers show the observed minimum and maximum.

Results. Figure 4 reports how often each method achieves the lowest test MSE among those satisfying a certain DPVar budget. ITD has the highest mean win rate at all three budgets. FBO ranks second under the Tight and Loose budgets, whereas GDP* narrowly surpasses FBO under the Mid budget. This performance is obtained despite FBO and ITD using only `OUT` labels for accuracy optimization, whereas the baselines use pooled `IN`+`OUT` labels. Overall, the two bilevel methods achieve the strongest win rates, particularly under the Tight constraint, although the seed-level results show variability.

4.3 Observed demographic attributes on Communities & Crime

Task and protocol. We complement the constructed-attribute benchmark with the Communities & Crime regression dataset, whose 1,994 rows represent US communities (Redmond, 2002). The outcome is standardized reported violent crime per capita. The sensitive attribute A is the observed ten-dimensional continuous demographic-composition profile: four race/ethnicity percentages, four age-composition percentages, the foreign-born percentage, and the percentage reporting limited English proficiency. We remove these coordinates and directly race-/immigration-conditioned fields from X, leaving 72 predictors. Across five split/model seeds, each split has 600 `IN`, 600 `OUT`, 397 `VAL`, and 397 `TEST` communities; DPVar evaluation uses five cross-fitting folds and three shared restarts. GDP*'s train-only two-component projection explains 63.7% of A's variance on average, but all methods are evaluated on the original ten-dimensional profile. Full protocol details are in Section G.

Results. Figure 5 gives the test pareto front frontier after validation selection. FBO, ITD and HSIC trace the lower envelope over most of the DPVar range. The R^2 penalty reaches the smallest absolute DPVar at a much larger accuracy cost; GDP* is competitive near the unconstrained end but does not attain the tight range. At very loose budgets, the methods converge to similar MSE. Validation-frozen paired comparisons with GDP*, uncertainty, additional fairness diagnostics, and runtime are deferred to Section G.

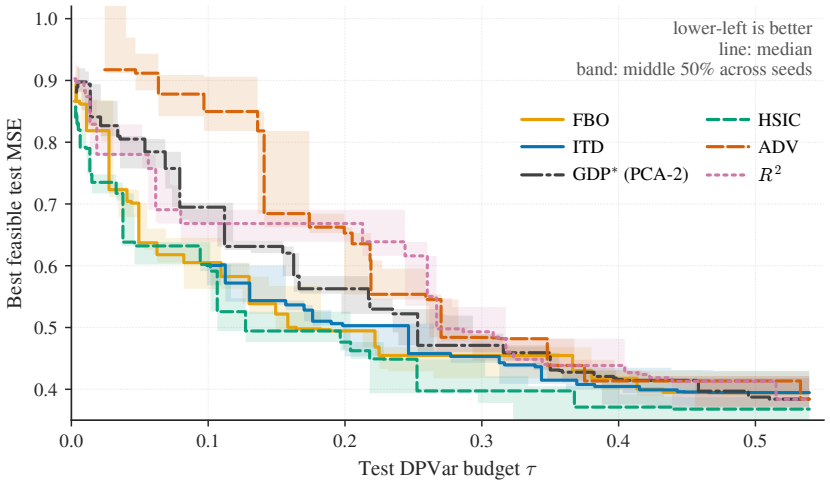


Figure 5: **Test accuracy–DPVar pareto front on Communities & Crime data with the ten-dimensional demographic profile.** Lines show the median best-feasible test MSE across five seeds; shaded bands show the 25th–75th percentile range. Lower-left is better.

5 Concluding Remarks

We studied how to train nonlinear regression models so that their average prediction changes less with a continuous, vector-valued sensitive attribute. We measure this change with DPVar. DPVar is a fairness metric that is difficult to compute, since it requires estimating an unknown conditional mean that changes with the predictor during training. We handle this difficulty by specializing the functional bilevel framework of Petrulionyte et al. (2024). For squared loss, the resulting population gradient has a closed form, which leads to the FBO update without a Hessian solve. We also study ITD, which differentiates through a finite sequence of updates to the conditional-mean model.

To test the methods across many datasets, we built a semi-synthetic benchmark from a fixed list of public regression tasks. The rows used to construct each task are separate from those used to train and evaluate the fairness methods. We estimate DPVar by fitting the conditional-mean model on one part of an evaluation split and measuring it on another, then exchanging the two parts and averaging the results. We further evaluated an observed ten-dimensional demographic-composition attribute on Communities & Crime, where FBO and ITD trace competitive accuracy–mean-parity trade-offs across five split/model seeds.

Scope and limitations. DPVar is a mean-parity criterion and should not be interpreted as a surrogate for full demographic parity in applications where distributional equality is required: it can leave higher-order conditional disparities – variance, tails, or richer distributional discrepancies – uncontrolled. Both training and evaluation also require estimating a conditional-mean function of A; when A is high-dimensional, this estimation problem can itself be statistically challenging. Finally, our tabular benchmark is semi-synthetic: the selected sensitive coordinates are used to create controlled tasks, not to make application-specific fairness claims about the original datasets. Communities & Crime supplies observed demographic composition, but its units are communities, and its outcome is reported crime; our study is therefore a contextual methodological audit, not an individual-level or causal fairness claim.

Broader impact. This work develops methodology for controlling mean disparities in predictions across continuous, high-dimensional sensitive attributes, with potential applications in domains such as credit scoring, insurance pricing, hiring, and recommendation, where average disparities across demographic profiles are a primary fairness concern. By making such control compatible with neural-network predictors and with sensitive attributes that are themselves continuous or vector-valued, the framework can reduce reliance on coarse demographic groupings that may misrepresent affected populations. At the same time, DPVar is a mean-parity criterion: it does not guarantee distributional fairness, equality of opportunity, or individual fairness, and a low DPVar value should not be interpreted as a certificate of overall fairness, which admits context-dependent definitions.

AI USE STATEMENT

In this work, we used generative AI tools to assist with drafting and editing text, creating or revising scientific visualizations, implementing and reviewing standard experimental code, explaining technical concepts, and formulating or checking mathematical statements. We did not use generative AI tools for translation or for cleaning and reformatting datasets; qualitative and thematic data analysis is not applicable to this work. The authors reviewed all AI-assisted material, checked mathematical claims and citations, tested AI-assisted code, and verified reported results against experiment artifacts. We take responsibility for the final content of this work, including text, claims, code and figures.

REPRODUCIBILITY STATEMENT

Section 3 defines the objective and estimators; Appendices A and C give the algorithms, assumptions, and proof. Section 4 and Appendices E–G document the data, preprocessing, splits, seeds, budgets, grids, selection, and evaluation. The anonymous supplement provides implementations, dependency specifications, and launch and analysis scripts for the tabular benchmark and Communities & Crime experiments, together with aggregate inputs for reproducing figures. Appendix E specifies the controlled synthetic experiment.

REFERENCES

Alekh Agarwal, Alina Beygelzimer, Miroslav Dudik, John Langford, and Hanna Wallach. A reductions approach to fair classification. *Proceedings of Machine Learning Research (PMLR)*, 80: 60–69, 2018.

Solon Barocas and Andrew D. Selbst. Big data's disparate impact. *California Law Review*, 104(3): 671–732, 2016.

Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning: Limitations and Opportunities*. MIT Press, 2023.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. Mutual information neural estimation. *Proceedings of Machine Learning Research (PMLR)*, 80:531–540, 2018.

Arthur Charpentier, François Hu, and Philipp Ratz. Mitigating discrimination in insurance with wasserstein barycenters. *PKDD/ECML Workshops*, 2023.

Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big Data*, 5(2):153–163, 2017.

Evgenii Chzhen and Nicolas Schreuder. A minimax framework for quantifying risk-fairness trade-off in regression. *The Annals of Statistics*, 50(4):2416–2442, 2022.

Evgenii Chzhen, Christophe Denis, Mohamed Hebiri, Luca Oneto, and Massimiliano Pontil. Fair regression with Wasserstein barycenters. In *Advances in Neural Information Processing Systems (NIPS)*, 2020.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2019.

Harrison Edwards and Amos Storkey. Censoring representations with an adversary. *International Conference on Learning Representations (ICLR)*, 2016.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research (JMLR)*, 17(59):1–35, 2016.

Solenne Gaucher, Nicolas Schreuder, and Evgenii Chzhen. Fair learning with Wasserstein barycenters for non-decomposable performance measures. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2023.

Paul Glasserman and Mike Li. Should bank stress tests be fair? *Management Science*, 71(1):262–278, 2024. doi: 10.1287/mnsc.2022.02060.

Vincent Grari, Sylvain Lamprier, and Marcin Detyniecki. Fairness-aware neural Rényi minimization for continuous features. *International Joint Conference on Artificial Intelligence (IJCAI)*, 2020.

Arthur Gretton, Kenji Fukumizu, Choon H. Teo, Le Song, Bernhard Schölkopf, and Alex J. Smola. A kernel statistical test of independence. *Advances in Neural Information Processing Systems (NIPS)*, 2008.

Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in Neural Information Processing Systems (NIPS)*, 2016.

Zhimeng Jiang, Xiaotian Han, Chao Fan, Fan Yang, Ali Mostafavi, and Xia Hu. Generalized demographic parity for group fairness. *International Conference on Learning Representations (ICLR)*, 2022.

Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Fairness-aware classifier with prejudice remover regularizer. *Machine Learning and Knowledge Discovery in Databases*, pp. 35–50, 2012.

Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. In *Innovations in Theoretical Computer Science (ITCS 2017)*, volume 67 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 43:1–43:23, 2017.

Junpei Komiyama, Akiko Takeda, Junya Honda, and Hajime Shimao. Nonconvex optimization for regression with fairness constraints. In *Proc. International Conference on Machine Learning (ICML)*, 2018.

Matt J. Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

Thibaut Le Gouic, Jean-Michel Loubes, and Philippe Rigollet. Projection to fairness in statistical learning. *arXiv preprint 2005.11720*, 2020.

Zhu Li, Adrián Pérez-Suay, Gustau Camps-Valls, and Dino Sejdinovic. Kernel dependence regularizers and Gaussian processes with applications to algorithmic fairness. *Pattern Recognition*, 132: 108922, 2022.

Jeremie Mary, Clément Calauzènes, and Noureddine El Karoui. Fairness-aware learning for continuous attributes and treatments. *Proceedings of Machine Learning Research (PMLR)*, 97:4382–4391, 2019.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys*, 54:1–35, 2021.

Mustafa Safa Ozdayi, Murat Kantarcioglu, and Rishabh Iyer. Fair machine learning under limited demographically labeled data. *Workshop on Socially Responsible Machine Learning (SRML)*, 2022.

Fabian Pedregosa. Hyperparameter optimization with approximate gradient. *Proceedings of Machine Learning Research (PMLR)*, 48:737–746, 2016.

Adrián Pérez-Suay, Valero Laparra, Gonzalo Mateo-García, Jordi Muñoz-Marí, Luis Gómez-Chova, and Gustau Camps-Valls. Fair kernel learning. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2017*, volume 10534 of *Lecture Notes in Computer Science*, pp. 339–355. Springer, 2017.

Ieva Petrulionyte, Julien Mairal, and Michael Arbel. Functional bilevel optimization for machine learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

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

Yuji Roh, Kangwook Lee, Steven Euijong Whang, and Changho Suh. Fairbatch: Batch selection for model fairness. *International Conference on Learning Representations (ICLR)*, 2021.

Veronika Shilova, Emmanuel Malherbe, Giovanni Palma, Laurent Risser, and Jean-Michel Loubes. Fairness-aware grouping for continuous sensitive variables: Application for debiasing face analysis with respect to skin tone. *Proceedings of the 28th European Conference on Artificial Intelligence (ECAI)*, 2025.

Sofiane Tanji, Samuel Vaiter, and Yassine Laguel. Fairness-informed pareto optimization : An efficient bilevel framework. *arXiv preprint arXiv:2601.13448*, 2026.

Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. Openml: Networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2):49–60, 2014.

Mariia Vladimirova, Eustache Diemert, and Federico Pavone. FairJob: A Real-World Dataset for Fairness in Online Systems. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Shaokui Wei, Jiayin Liu, Bing Li, and Hongyuan Zha. Mean parity fair regression in rkhs. In *International conference on artificial intelligence and statistics*, pp. 4602–4628. PMLR, 2023.

Mehdi Yazdani-Jahromi, Ali Khodabandeh Yalabadi, AmirArsalan Rajabi, Aida Tayebi, Ivan Garibay, and Ozlem Garibay. Fair Bilevel Neural Network (FairBiNN): On Balancing Fairness and Accuracy via Stackelberg Equilibrium. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P. Gummadi. Fairness constraints: Mechanisms for fair classification. *Proceedings of Machine Learning Research (PMLR)*, 54:962–970, 2017.

Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. *Proceedings of the Conference on AI, Ethics, and Society (AIES)*, pp. 335–340, 2018.

A TRAINING ALGORITHMS

Algorithm 2 EstimateConditionalMean

Input: predictor parameters ω; initial inner parameters ϕ_0; inner stepsize η_{in}; number of inner steps M; IN split
$\phi \leftarrow \phi_0$
for $m = 1, \dots, M$ **do**
    Compute the empirical inner objective

$$\hat{L}_{\text{in}}(\omega, \phi) = \frac{1}{|\texttt{IN}|} \sum_{(x_i, a_i) \in \texttt{IN}} \big(f_\omega(x_i) - h_\phi(a_i)\big)^2$$

    Update

$$\phi \leftarrow \phi - \eta_{\text{in}} \nabla_\phi \hat{L}_{\text{in}}(\omega, \phi)$$

end for
Return: ϕ

Algorithm 3 ComputeHypergradient

Input: predictor parameters ω; inner parameters ϕ; fairness weight α; hypergradient mode HG $\in \{\text{FBO}, \text{ITD}\}$; unrolling length K; inner stepsize η_{in}; IN split; OUT split
if HG = FBO **then**
    Compute

$$\hat{\mu} \leftarrow \frac{1}{|\texttt{OUT}|} \sum_{a_i \in \texttt{OUT}} h_\phi(a_i)$$

    Treat $h_\phi(a_i)$ and $\hat{\mu}$ as fixed when differentiating with respect to ω
    Compute the accuracy gradient

$$g_{\text{acc}} \leftarrow \nabla_\omega \left[\frac{1}{|\texttt{OUT}|} \sum_{(x_i, y_i) \in \texttt{OUT}} \big(f_\omega(x_i) - y_i\big)^2 \right]$$

    Compute the fairness gradient

$$g_{\text{fair}} \leftarrow \nabla_\omega \left[\frac{2\alpha}{|\texttt{IN}|} \sum_{(x_i, a_i) \in \texttt{IN}} \big(h_\phi(a_i) - \hat{\mu}\big) f_\omega(x_i) \right]$$

    Return: $g_{\text{acc}} + g_{\text{fair}}$
else
    $\tilde{\phi}^{(0)} \leftarrow \phi$
    for $k = 0, \dots, K-1$ **do**
        $\tilde{\phi}^{(k+1)} \leftarrow \tilde{\phi}^{(k)} - \eta_{\text{in}} \nabla_\phi \hat{L}_{\text{in}}(\omega, \tilde{\phi}^{(k)})$
    end for
    Define the unrolled empirical outer objective

$$\hat{L}^{\text{ITD}}_{\text{out}}(\omega) = \frac{1}{|\texttt{OUT}|} \sum_{(x_i, y_i) \in \texttt{OUT}} \big(f_\omega(x_i) - y_i\big)^2 + \alpha \, \widehat{\text{Var}}_{a_i \in \texttt{OUT}}\big(h_{\tilde{\phi}^{(K)}}(a_i)\big)$$

    Return: $\nabla_\omega \hat{L}^{\text{ITD}}_{\text{out}}(\omega)$
end if

B Additional Related Work

Demographic parity and dependence penalties. Demographic-parity methods often regularize training by penalizing dependence between the prediction $f_\omega(X)$ and the sensitive attribute A. This includes correlation/covariance penalties (denoted R^2) (Zafar et al., 2017), kernel-based dependence measures such as HSIC (Gretton et al., 2008), and mutual-information inspired objectives (Belghazi et al., 2018; Kamishima et al., 2012). These methods are attractive because they are single-level objectives that can be optimized with standard stochastic-gradient tools. Their limitation, in our setting, is that the optimized dependence penalty is not necessarily the demographic-parity criterion we care about. Depending on the kernels or adversary, it may target full dependence or selected joint moments rather than directly fitting and penalizing $a \mapsto \mathbb{E}[f_\omega(X) \mid A = a]$ (Mary et al., 2019; Jiang et al., 2022).

HSIC baseline and choice of kernel. Pérez-Suay et al. (2017) propose both a linear formulation (Section 2.2) and a kernel formulation (Section 2.3). For a linear predictor $p = X\beta$ and a sensitive matrix $A \in \mathbb{R}^{n \times k_A}$, their linear solution is

$$\widehat{\beta}_\mu = \left(X^\top X + \lambda I + \frac{\mu}{n^2} X^\top H A A^\top H X\right)^{-1} X^\top y, \qquad H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top. \tag{4}$$

This is HSIC with linear kernels on the scalar prediction and on the complete sensitive vector. It penalizes the squared cross-covariance between p and all columns of A simultaneously. The paper explicitly allows multidimensional sensitive variables. The authors' release confirms this interpretation:[3] `FL.m` forms `H*Q*Q'*H` and solves exactly the system in Equation (4), where `Q` is a matrix rather than a single sensitive column. The accompanying scripts use two indicator columns for sex in `FKL_sex.m`, and seven columns for race and sex in `FKL_sex_race.m`. They also report variants that remove these columns from the predictor while retaining the full `Q` in the fairness penalty, which matches our convention that A is available during training but excluded from X at prediction time.

Our tabular benchmark passes the full continuous A directly to this linear construction. Across the 21 datasets, k_A ranges from 3 to 20; we neither select one coordinate nor project A to a lower-dimensional representation. Because our tasks are regressions, we select the ridge parameter by validation MSE rather than the validation classification accuracy used by the released experiment script. We retain the release's seven-value ridge grid and 25-value fairness path, and add a zero-penalty anchor for comparison with the unconstrained model. Computationally, we use the exact identity

$$X^\top H A A^\top H X = (A^\top H X)^\top (A^\top H X). \tag{5}$$

Thus every training observation and every sensitive coordinate is retained, while no $n \times n$ sensitive Gram matrix is materialized. This is an algebraic rearrangement of Equation (4), not a sampling or kernel approximation.

The paper's nonlinear formulation and the released `FKL.m` instead replace the linear model in X by RBF kernel-ridge regression and replace the sensitive Gram matrix $AA^\top$ by an RBF Gram matrix on A; the HSIC kernel on the scalar fitted predictions remains linear. The RBF kernel on A can detect nonlinear dependence that the linear kernel misses, but `FKL.m` constructs sample-by-sample input and sensitive Gram matrices and solves an $n \times n$ kernel-ridge system. Our largest tasks have up to 66,666 pooled training observations, for which one dense Gram matrix already contains about 4.4 billion entries; computing it in blocks would reduce memory but not the quadratic pairwise work or the kernel-system cost. Random features, Nyström approximations, minibatch HSIC, or subsampling could make an RBF penalty feasible, but none is used in the authors' release and each would introduce an additional approximation whose quality must be tuned. We therefore use the authors' exact, scalable linear formulation rather than introduce a new approximate RBF variant. The trade-off is explicit: this baseline controls linear, second-order dependence between predictions and A, not arbitrary nonlinear dependence.

Adversarial debiasing. Adversarial training enforces invariance by training a predictor jointly with an adversary that attempts to recover A from intermediate representations or from $f_\omega(X)$ (Zhang

[3] Author release: https://www.uv.es/pesuaya/data/code/FairKernelLearning2017.zip.

et al., 2018; Edwards & Storkey, 2016; Ganin et al., 2016). This approach can target a rich family of dependence notions (depending on the adversary class) and has been successful empirically. At the same time, it can be sensitive to optimization hyperparameters and to the choice of adversary's architecture. Additionally, similarly to dependence penalties, adversarial debiasing does not directly estimate the conditional mean that defines DPVar (Jiang et al., 2022; Mary et al., 2019).

Scope of the CoD comparison. The coefficient-of-determination (CoD) method of Komiyama et al. (2018) handles multiple numeric sensitive attributes and includes nonlinear kernel and random-Fourier-feature extensions, so it is not restricted to linear regression. Its released estimator, however, includes the sensitive attribute in the prediction function and also uses it to residualize the other covariates at validation and test time; the released optimization further depends on Gurobi or Mosek for its QCQP/SDP problems.[4] This is a different hypothesis class from our common $f_\omega(X)$ protocol, in which A is removed from X and is unavailable to every predictor at inference. Removing the sensitive-input component or replacing the specialized solver with a neural penalty would therefore be a new adaptation rather than a reproduction. We instead include an R^2 baseline that applies the same multiple-linear-predictability statistic as a differentiable penalty to the common neural predictor; we do not present it as the original constrained CoD solver.

Scope of Wasserstein post-processing. Fair-regression methods based on Wasserstein barycenters post-process an unconstrained regressor by mapping each finite sensitive group's conditional output distribution to a common barycenter (Chzhen et al., 2020; Le Gouic et al., 2020; Chzhen & Schreuder, 2022). Their prediction rule uses group membership at inference; the released implementation of Chzhen et al. (2020) specializes the construction to two groups.[5] With multidimensional continuous A, nearly every observed value defines its own group, so this empirical group-quantile map is not defined without first imposing a discretization or clustering rule. Such a rule would change both the method and its full-distribution fairness target, and would again require A at inference. We therefore report these methods as out of scope rather than introducing an unvalidated discretized variant.

Bilevel fairness methods. Our use of bilevel optimization differs from recent bilevel approaches to fair learning (Roh et al., 2021; Ozdayi et al., 2022; Yazdani-Jahromi et al., 2024; Tanji et al., 2026). Earlier bilevel fairness methods use the outer problem to adapt training mechanisms such as minibatch composition or sample weights: FairBatch (Roh et al., 2021) adjusts group-dependent batch-selection parameters to improve standard group-fairness metrics, while BiFair (Ozdayi et al., 2022) uses bilevel optimization for fair learning when demographic labels are available only for a small subset of the data. These methods are bilevel in the training procedure, but their inner problems do not estimate the conditional-mean function defining a fairness criterion. FairBiNN (Yazdani-Jahromi et al., 2024) takes an architecture-centered view: the model is split into accuracy-responsible and fairness-responsible components, and the bilevel structure comes from optimizing these two sets of layers separately. In this sense, FairBiNN is related to our work as a bilevel fairness method, which is why it appears in Figure 2. However, it differs in two key ways: (i) it targets standard group-fairness losses for discrete sensitive groups rather than the conditional-mean DPVar criterion, and (ii) its accuracy-fairness trade-off is controlled through architectural and optimization choices, such as the placement of fairness layers and the relative learning rates of the two players, rather than by a penalty parameter weighting the DPVar criterion. The bilevel Pareto framework of Tanji et al. (2026) is also distinct. Their BADR method is motivated by the observation that standard fairness regularization can produce models that are Pareto-inefficient across predefined groups. In contrast to our approach, the bilevel structure in BADR is not induced by an unknown conditional-mean function. Instead, it is used to select a Pareto-efficient fair model among predefined groups. Thus, existing bilevel fairness methods use bilevel optimization to adapt the training procedure, structure the architecture, or select a point on a group-risk Pareto frontier; in our work, the bilevel structure is induced by the DPVar objective itself, through the estimation of $h^\star_\omega(a) = \mathbb{E}[f_\omega(X) \mid A = a]$. The general functional bilevel and adjoint framework was introduced by Petrulionyte et al. (2024). Here we specialize it to DPVar: the squared inner loss gives a closed-form population adjoint, from which we construct the plug-in update studied in this paper.

[4] Official implementation: https://github.com/jkomiyama/fairregresion.

[5] Official implementation: https://github.com/lucaoneto/NIPS2020_Fairness.

C TOTAL-GRADIENT DERIVATION FOR THE SQUARED-LOSS BILEVEL OBJECTIVE

Proof of Proposition 3.1. Assume that the IN and OUT laws have the same marginal distribution for A, and that the required moments exist so that differentiation and expectation can be exchanged below. Let P_{out} be the law on (X, Y) and $P_{A_{\text{out}}}$ the marginal on A. The outer objective can be written as:

$$L_{\text{out}}(\omega, h) = \underbrace{\mathbb{E}_{(X,Y)\sim P_{\text{out}}}\left[(f_\omega(X) - Y)^2\right]}_{\text{accuracy}} + \alpha \underbrace{\mathbb{E}_{A\sim P_{A_{\text{out}}}}\left[(h(A) - \mu)^2\right]}_{\text{DP variance penalty}},$$
$$\mu = \mathbb{E}_{A\sim P_{A_{\text{out}}}}\left[h(A)\right].$$

1) Functional gradient in h. We note $h(A)$, $\varphi(A)$, and $\mathbb{E}_{A\sim P_{A_{\text{out}}}}$ as h, φ, and $\mathbb{E}$ for readability. We work in $L_2(P_{A_{\text{out}}})$ the space of square integrable functions wrt. the measure $P_{A_{\text{out}}}$ equipped with the inner product $\langle u, v\rangle_{L_2(P_{A_{\text{out}}})} := \mathbb{E}_{A\sim P_{A_{\text{out}}}}[u(A)v(A)]$. For any direction $\varphi \in L_2(P_{A_{\text{out}}})$, consider the perturbation $h + \varepsilon\varphi$ and define:

$$\mu_\varepsilon := \mathbb{E}[h + \varepsilon\varphi] = \mathbb{E}\left[h\right] + \varepsilon\mathbb{E}[\varphi] = \mu + \varepsilon\mathbb{E}[\varphi].$$

Then consider the perturbed DP variance penalty:

$$\begin{aligned}
\mathbb{E}\left[(h + \varepsilon\varphi - \mu_\varepsilon)^2\right] &= \mathbb{E}\left[(h + \varepsilon\varphi - (\mu + \varepsilon\mathbb{E}[\varphi]))^2\right] \\
&= \mathbb{E}\left[(h - \mu + \varepsilon\left(\varphi - \mathbb{E}[\varphi]\right))^2\right] \\
&= \mathbb{E}\left[(h - \mu)^2\right] + 2\varepsilon\mathbb{E}\left[(h - \mu)\left(\varphi - \mathbb{E}[\varphi]\right)\right] + \varepsilon^2\mathbb{E}\left[\left(\varphi - \mathbb{E}[\varphi]\right)^2\right].
\end{aligned}$$

Keeping only the first-order term in ε gives the differential

$$\partial_h\mathbb{E}\left[(h - \mu)^2\right][\varphi] = 2\mathbb{E}\left[(h - \mu)\left(\varphi - \mathbb{E}[\varphi]\right)\right] = 2\mathbb{E}\left[(h - \mu)\varphi\right],$$

since $\mathbb{E}[h - \mu] = \mathbb{E}[h] - \mu = \mu - \mu = 0$. Assuming the IN and OUT splits are drawn i.i.d. from the same population, we have $P_{A_{\text{in}}} = P_{A_{\text{out}}}$, so for any $\varphi \in L_2(P_{A_{\text{in}}})$,

$$\begin{aligned}
\partial_h L_{\text{out}}(\omega, h)[\varphi] &= 2\alpha\mathbb{E}_{A\sim P_{A_{\text{out}}}}\left[(h(A) - \mu)\varphi(A)\right] \\
&= 2\alpha\mathbb{E}_{A\sim P_{A_{\text{in}}}}\left[(h(A) - \mu)\varphi(A)\right].
\end{aligned}$$

2) Adjoint. The inner objective is $L_{\text{in}}(\omega, h) = \mathbb{E}_{(X,A)\sim P_{\text{in}}}\left[(f_\omega(X) - h(A))^2\right]$, whose Hessian w.r.t. h is $\partial^2_{hh}L_{\text{in}} = 2I$ on $L_2(P_{A_{\text{in}}})$. The adjoint $a^\star_\omega \in L_2(P_{A_{\text{in}}})$ solves

$$\partial^2_{hh}L_{\text{in}}(\omega, h^\star_\omega)a^\star_\omega = -\partial_h L_{\text{out}}(\omega, h^\star_\omega),$$

hence

$$\begin{aligned}
2a^\star_\omega(a) &= -2\alpha(h^\star_\omega(a) - \mu_\omega) \\
a^\star_\omega(a) &= -\alpha(h^\star_\omega(a) - \mu_\omega), \qquad \mu_\omega := \mathbb{E}_{A\sim P_{A_{\text{out}}}}[h^\star_\omega(A)].
\end{aligned}$$

3) Total gradient By the functional adjoint identity,

$$\nabla_\omega F(\omega) = \partial_\omega L_{\text{out}}(\omega, h^\star_\omega) + \partial^2_{\omega h}L_{\text{in}}(\omega, h^\star_\omega)[a^\star_\omega].$$

Compute each term

- **Direct outer term:**

$$\partial_\omega L_{\text{out}}(\omega, h^\star_\omega) = 2\mathbb{E}_{\text{out}}\left[(f_\omega(X) - Y)\partial_\omega f_\omega(X)\right].$$

- **Implicit (adjoint) term:**

$$\partial_h L_{\text{in}}(\omega, h)[\varphi] = -2\mathbb{E}_{\text{in}}\left[(f_\omega(X) - h(A))\varphi(A)\right],$$

differentiating w.r.t. ω:

$$\partial^2_{\omega h}L_{\text{in}}(\omega, h)[a] = -2\mathbb{E}_{\text{in}}\left[a(A)\partial_\omega f_\omega(X)\right],$$

writing the full implicit term:

$$\partial^2_{\omega h}L_{\text{in}}(\omega, h^\star_\omega)[a^\star_\omega] = -2\mathbb{E}_{\text{in}}\left[a^\star_\omega(A)\partial_\omega f_\omega(X)\right] = 2\alpha\mathbb{E}_{\text{in}}\left[\left(h^\star_\omega(A) - \mu_\omega\right)\partial_\omega f_\omega(X)\right].$$

Putting it together,

$$\nabla_\omega F(\omega) = 2\mathbb{E}_{X,Y\sim P_{\text{out}}}\left[(f_\omega(X)-Y)\partial_\omega f_\omega(X)\right] + 2\alpha\mathbb{E}_{A,X\sim P_{\text{in}}}\left[\left(h^\star_\omega(A)-\mu_\omega\right)\partial_\omega f_\omega(X)\right],$$

$$\mu_\omega = \mathbb{E}_{A\sim P_{A_{\text{out}}}}[h^\star_\omega(A)].$$

□

D Example: Job Recommendation.

Consider job recommendation in online advertising, where some demographic profiles should not systematically receive lower-salary recommendations on average. This is the setting considered in the FairJob benchmark (Vladimirova et al., 2024). In this benchmark, the protected attribute A is not directly observed; instead, the released dataset provides a binary gender proxy inferred from users' behavioral interactions. This binary proxy makes the dataset compatible with standard demographic-parity methods, which are usually designed for discrete sensitive groups and do not directly apply to high-dimensional continuous sensitive attributes. At the same time, it highlights a limitation of this approach: a richer behavioral signal must be reduced to a hard demographic label before these methods can be used. Our method avoids this reduction when the richer sensitive representation is available. Rather than collapsing the sensitive signal to a binary class and enforcing independence from that class, it can work directly with a richer representation and control whether predicted salaries or recommendation scores are systematically higher or lower on average across demographic profiles. Moreover, because our approach controls the conditional expectation rather than enforcing full independence, it does not require the entire distribution of recommendations to be identical across all demographic profiles. This is important in job recommendation, where different groups may legitimately differ in preferences or constraints, such as desired flexibility, remote-work options, commuting tolerance, or schedule compatibility. Enforcing full independence risks washing out this structure and producing recommendations that are less useful to everyone. By contrast, controlling the conditional expectation means that demographic profiles receive comparable salary levels on average, while still allowing recommendations to reflect heterogeneous needs.

E Controlled Synthetic Experiment

Let $U, V, W, Z, \xi, \varepsilon \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$ and define

$$A = \left(\frac{U+\kappa W}{\sqrt{1+\kappa^2}}, \frac{U-\kappa W}{\sqrt{1+\kappa^2}}, V\right), \quad q(W) = \frac{\sin(2W) - 2e^{-2}W}{c_q},$$

$$X_1 = q(W) + \sigma_x\xi, \qquad X_2 = (1+\rho U)Z, \qquad Y = q(W) + \beta X_2 + \sigma_y\varepsilon,$$

where $c_q^2 = (1-e^{-8})/2 - 4e^{-4}$ and $(\kappa, \rho, \beta, \sigma_x, \sigma_y) = (.35, .5, 2, .15, .1)$. The first two coordinates of A determine U and W. The normalization gives $\mathrm{Var}(q(W)) = 1$, while the subtraction makes $q(W)$ uncorrelated with W. Thus X_1 contains a nonlinear mean disparity that a linear dependence penalty can miss. In contrast, $\mathbb{E}[X_2 \mid A] = 0$, although $\mathrm{Var}(X_2 \mid A) = (1+\rho U)^2$ changes with A; penalizing that change is unnecessary for mean parity.

For the linear reference $f_b(X) = b_1X_1 + b_2X_2$,

$$\mathbb{E}[f_b(X) \mid A] = b_1 q(W), \qquad \mathrm{DPVar}(f_b) = b_1^2,$$
$$\mathrm{MSE}(f_b) = (b_1-1)^2 + \sigma_x^2 b_1^2 + (1+\rho^2)(b_2-\beta)^2 + \sigma_y^2.$$

Consequently, minimizing $\mathrm{MSE} + \alpha\,\mathrm{DPVar}$ gives $b_1 = (1+\sigma_x^2+\alpha)^{-1}$ and $b_2 = \beta$. This curve is the dashed population reference in Figure 3.

Each seed draws four independent splits of 2,000 observations. Every method trains one model at zero penalty and 16 logarithmically spaced positive penalties: $[10^{-3}, 10^2]$ for FBO and adversarial debiasing, $[10^{-2}, 10^3]$ for ITD, $[10^{-4}, 10^2]$ for GDP* and R^2, and $[10^{-2}, 10^4]$ for HSIC. For the simulator-based test metric, we hold each of the 2,000 test values of A fixed, resample (Z, ξ) 128 times, average the resulting predictions, and take the variance across A, subtracting the finite-draw Monte Carlo contribution. The same draws are shared across methods within a seed. All 17 trained models are reported; no test quantity is used for selection.

F TABULAR BENCHMARK PROTOCOL

Task roster and feature types. We begin with a versioned roster of 105 OpenML and UCI regression tasks. The prepared artifacts record the source dataset and target, feature names and positions, semantic types, missing values, and checksums. The benchmark is numeric-only: categorical, Boolean, date/time, and identifier-like columns are excluded. Only source-continuous columns can be selected for A; discrete numeric columns may remain in X.

Separating construction from evaluation. We use stable source-row identifiers and a deterministic hash to divide each task into a construction pool and a disjoint experiment pool. The construction pool targets 20% of the complete rows, with a minimum of 2,000 and a cap of 20,000, and is used for every target-informed decision described below. It is split again into `FIT`, `DIAGNOSTIC`, and `EVALUATION` subsets in proportions $1/2$, $1/4$, and $1/4$. No construction row is used to train, validate, or test a fairness method. After withholding this pool, a task must retain at least 5,000 complete experiment rows.

Constructing A. On construction `FIT`, we first identify continuous, nonconstant features with sufficient numerical variation. Candidate features are screened by their absolute correlation with the target. We then score each candidate using both its target correlation and a proxy score measuring its correlation with other available features. This favors coordinates that matter for prediction but whose information is not isolated to a single input column. The highest-scoring quarter forms A, with a minimum of three coordinates; at least one coordinate must remain in X. The selected coordinates are removed from X in every subsequent experiment.

This construction is designed to make the benchmark informative rather than artificially easy. If A were unrelated to the target, enforcing fairness would have little effect on accuracy. If its information disappeared when its columns were removed, a predictor using only X would have little disparity to mitigate. Selecting target-associated coordinates with proxies in the remaining features creates a measurable accuracy–fairness trade-off across datasets. The selected coordinates are synthetic sensitive attributes for this stress test; we do not claim that they are sensitive in their source applications.

Dataset eligibility. We check the construction on held-out rows before any fairness method is trained. A task is retained only if (i) a reference predictor using X achieves $R^2 \geq .10$, (ii) A explains at least 10% of the variance in its predictions, and (iii) removing the estimated A-dependent component increases MSE by at least 2% of the target variance. These checks rule out tasks with no useful predictive signal, no measurable disparity, or no meaningful accuracy–fairness tension. Structural rules also remove duplicates and tasks requiring temporal, grouped, or survival-specific evaluation. Applying the fixed rules to the 105-task roster yields the 21 benchmark datasets. This set is frozen before the fairness methods are run; validation ranges and test results cannot change benchmark membership.

Final experiment splits and preprocessing. For each final run, we use at most 100,000 rows from the experiment pool and allocate them in proportions $1/3$, $1/3$, $1/6$, and $1/6$ to `IN`, `OUT`, `VAL`, and `TEST`. Centering and scaling are fitted on `IN`+`OUT` and applied unchanged to `VAL` and `TEST`. All experiment-pool rows are complete, so the defensive imputation code does not alter the data. The benchmark manifest records the selected feature names and indices, row-partition rule, and construction cutoff.

Hyperparameter-search budgets. The gradient-trained methods use matched search budgets. FBO, ITD, R^2, adversarial debiasing, and GDP* each evaluate eight method-specific hyperparameter settings for every positive penalty. All five methods start from $\{0, .5, 2, 10, 50, 200, 1000, 5000\}$ and use the same validation-driven refinement rule, with at most twelve distinct penalties. At zero penalty, inactive fairness hyperparameters are collapsed to two learning-rate choices; hence each method evaluates at most $2 + 11 \times 8 = 90$ configurations. Every gradient-trained configuration uses 300 predictor updates and the same cross-fitted validation DPVar diagnostic.

HSIC follows a different, method-specific protocol because it has a released closed-form solution rather than a gradient-trained neural objective. We select its ridge once by validation MSE from the released seven-value grid, freeze that choice, and evaluate the released 25-value fairness path together

with a zero-penalty anchor. Thus the gradient-trained methods receive the same maximum number of tuning opportunities, while HSIC follows its published search path. Per-configuration runtimes are not matched because the methods have different computational costs.

G Communities & Crime Experiment

Data and sensitive profile. The Communities & Crime dataset combines community-level demographic and socioeconomic measurements with reported violent crime for 1,994 US communities (Redmond, 2002). We predict the standardized `ViolentCrimesPerPop` outcome. The target-independent sensitive profile is

$$A = (\texttt{racepctblack}, \texttt{racePctWhite}, \texttt{racePctAsian}, \texttt{racePctHisp}, \texttt{agePct12t21}, \texttt{agePct12t29}, \texttt{agePct16t24}, \texttt{agePct65up}, \texttt{PctForeignBorn}, \texttt{PctNotSpeakEnglWell}) \in \mathbb{R}^{10}.$$

We remove these coordinates, identifiers, race-conditioned income and police fields, immigration/language-composition fields, and the highly incomplete police block from the predictor input, leaving the same 72 predictors in every run. Missing-value imputation and standardization are fit on `IN`+`OUT`. Each target-stratified split contains 600 `IN`, 600 `OUT`, 397 `VAL`, and 397 `TEST` communities.

Training protocol. Every method trains exactly 32 distinct models per seed. For FBO, ITD, R^2, adversarial debiasing, and GDP*, we select 32 configurations from a 13-value penalty grid and the method-specific settings used in the tabular benchmark. This subset is fixed before any validation metric is observed, covers every penalty value, and retains both predictor learning rates at zero penalty. Each configuration uses 500 predictor updates. HSIC follows the search procedure in its released implementation rather than the neural-method grid: seven ridge candidates are compared by validation MSE, the selected zero-penalty model is reused, and 25 additional models are fitted along the fairness path. This also gives exactly 32 fits. The number of fitted models is therefore identical across methods.

Validation DPVar is estimated with five folds, three shared restarts, and 300 fitting steps. Only the union of configurations chosen by validation scalarizations and the validation Pareto set is evaluated on `TEST`, where the same five-fold, three-restart diagnostic is refit for 1,000 steps.

Table 1: Mean paired difference $\Delta_w = (\mathrm{MSE} + w\,\mathrm{DPVar})_m - (\mathrm{MSE} + w\,\mathrm{DPVar})_{\mathrm{GDP}^*}$ on Communities & Crime. Negative values favor the DPVar-targeting method; parentheses give the number of five seeds with a negative difference. Configurations are selected on validation only.

Method	$w = 0$	$w = .5$	$w = 2$	$w = 10$
ITD–GDP*	$+.011$ (1/5)	$-.023$ (4/5)	$-.124$ (5/5)	$+.641$ (0/5)
FBO–GDP*	$+.010$ (2/5)	$-.036$ (4/5)	$-.143$ (5/5)	$-.025$ (4/5)

At $w = 2$, FBO lowers mean MSE by .0202 and DPVar by .0616, lowering the common test objective by .1434 in all five seeds; its pointwise conditional interval is $[-.181, -.110]$. At $w = 10$, the corresponding changes are $-.0071$, $-.0018$, and $-.0254$ in four seeds, with interval $[-.049, .002]$. These intervals resample the 44 observed source-state clusters with shared draws across methods and seeds. Because the five test splits reuse one finite dataset, they are not five independent datasets; even five same-sign pairs give an exact two-sided sign-test $p = .0625$. We therefore interpret the pattern as consistency across fitted pipelines, not as a confirmatory significance claim.

As an estimator check, an independently cross-fitted ExtraTrees regressor yields DPVar values closely aligned with the neural diagnostic across 60 distinct selected evaluations (Pearson $r = .999$, Spearman $\rho = .995$). Median training time over five seeds is .58 seconds for the full HSIC search, 9.40 seconds for FBO, 46.57 seconds for ITD, and 111.32 seconds for GDP*. At $w = 10$, ITD selects its maximum penalty in all five seeds, FBO in three, and GDP* in four, so the strong-weight comparisons should be read with this boundary diagnostic in mind.

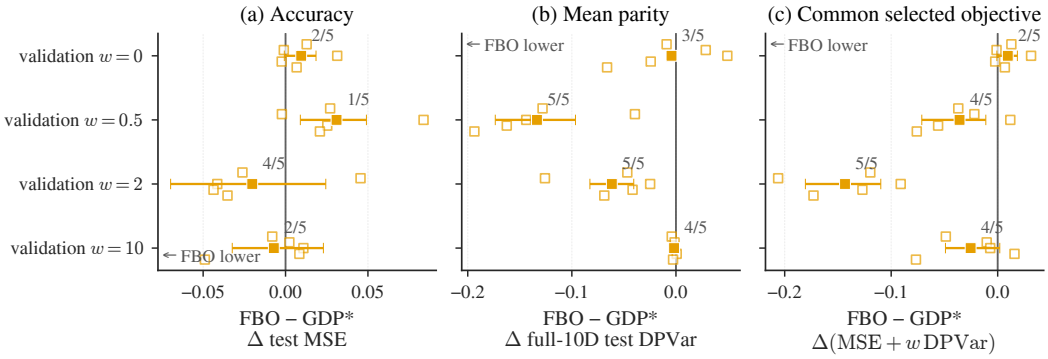


Figure 6: **Test paired FBO–GDP* comparison.** Each row uses the configuration minimizing the corresponding validation objective before test evaluation. Open squares are the five split/model seeds; the filled square and bar give the mean and pointwise 95% source-state-cluster interval conditional on the five fitted pipelines. Negative values favor FBO. Panel (c) uses the same weight w as validation selection. The labels count seeds favoring FBO.