CAUSAL REGRESSION: LEARNING CAUSAL MECHANISMS FOR ROBUST AND INTERPRETABLE PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Standard regression models can be vulnerable to corrupted training labels. This paper develops Causal Regression, a framework for structuring prediction around a latent individual representation and a mechanism that maps this representation to an outcome. We introduce CausalEngine, a neural implementation grounded in the Distribution-consistency Structural Causal Model (DiscoSCM). Its Cauchy parameterization and linear mechanism permit analytical uncertainty propagation and end-to-end optimization without sampling-based inference. The architecture also separates uncertainty associated with incomplete knowledge of an individual from residual environmental variation.

This revised version corrects the implementation and re-evaluates the empirical claims. The historical early-stopping code stored shallow, tensor-aliased checkpoints and therefore did not retain an independent validation-best model. With deep-copied checkpoint restoration, several baselines improve and the previously reported CausalEngine margins become smaller. Across five paired seeds, the fixed standard mode retains a positive label-permutation result on both Synthetic generators and six of eight public datasets, but it does not support uniform superiority across corruption types: on California Housing it wins all shuffle comparisons and none of the asymmetric, outlier, or systematic comparisons. We report fixed-primary and best-of-family analyses separately and limit the conclusion to predictive robustness under the tested corruption protocols.[1]

1 INTRODUCTION

Regression is usually presented as a problem of predicting an outcome from a set of observed features (Hastie et al., 2009; Bishop, 2006). Causal inference, by contrast, is often organized around treatments and interventions (Pearl, 2009; Neyman, 1923; Rubin, 1974). This paper asks whether a causally structured model can still be useful in an ordinary prediction task, such as California Housing, where no treatment is specified.

Our starting point is to distinguish the observed predictors X from a latent individual representation U. In the Causal Regression model, X provides information about U, and the outcome is generated through a mechanism

$$Y = f(U, \varepsilon). \tag{1}$$

Here ε represents residual environmental variation. The separation is motivated by the Distribution-consistency Structural Causal Model (DiscoSCM) (Gong et al., 2024), which distinguishes stable individual variation from event-level noise.

This parameterization should not be read as automatic causal identification. From observational prediction data alone, a learned U need not equal a unique or true latent cause, and predictive robustness does not validate interventions or counterfactuals. We instead study a narrower question: does explicitly factorizing representation and residual variation change how prediction degrades when the training feature–label relation is corrupted?

We instantiate Causal Regression with **CausalEngine**, a four-stage architecture (Figure 1):

1. **Perception**: maps input features X to a representation Z;

[1] Code and corrected result manifests will accompany the new submission.

2. **Abduction**: infers a distribution over the individual representation U from Z;
3. **Action**: maps U to a scalar score S through a linear mechanism;
4. **Decision**: maps S to the task-specific prediction Y.

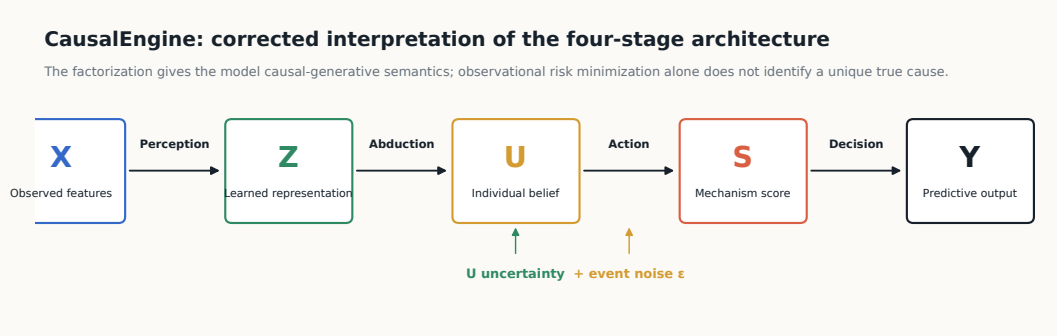

Figure 1: CausalEngine maps features to predictions through Perception, Abduction, Action, and Decision. The stage names describe the model factorization and distinguish individual uncertainty from event noise; causal identification requires assumptions beyond the diagram.

Two technical choices make this architecture analytically tractable. First, the Action stage is linear in the scalar score. This follows the practical hypothesis that a sufficiently expressive representation can linearize a downstream decision, as is common in representation learning and reward modeling (Ouyang et al., 2022; Zhong et al., 2025). Second, the Abduction distribution is Cauchy. Its stability under affine transformations supports closed-form propagation, while its heavy tails provide a robust alternative to Gaussian residual modeling. We use the Cauchy distribution for these mathematical and statistical properties, not as a general claim that one distribution is uniquely "causal."

The staged architecture also provides an interpretable parameterization: the model exposes intermediate representations and separately parameterizes uncertainty associated with incomplete individual information and residual variation. This is more structured than a single undifferentiated predictive variance, but it does not guarantee that each internal quantity has a uniquely identified real-world meaning. Direct interpretability claims therefore require calibration and semantic validation in addition to inspection of the architecture.

This revised paper makes three contributions:

1. **Causal Regression framework.** We formulate ordinary regression through a latent individual representation and an outcome mechanism, giving a causal-modeling perspective on prediction tasks without an explicit treatment.
2. **Tractable CausalEngine architecture.** We develop a four-stage implementation whose linear mechanism and Cauchy parameterization permit analytical uncertainty propagation and end-to-end learning.
3. **Corrected robustness evaluation.** We disclose and repair a checkpoint-aliasing defect in the historical implementation, separate a fixed primary mode from best-of-family selection, and re-evaluate label-corruption robustness with explicit seeds, provenance, and paired uncertainty. The corrected evidence supports corruption-specific rather than universal robustness.

The remainder of the paper presents the Causal Regression framework, CausalEngine, the corrected experiments, and their limitations.

1.1 Related Work

Structural causal models and latent representations. Deep Structural Causal Models use flexible generative models to represent causal mechanisms and counterfactuals (Pawlowski et al., 2020; Xia et al., 2021; Poinsot et al., 2024). Their causal interpretation depends on structural assumptions such as a known graph, identifiable mechanisms, or restrictions on confounding. Causal discovery

methods likewise require assumptions including faithfulness or causal sufficiency and can be unstable in finite samples (Spirtes et al., 2000; Glymour et al., 2019). Causal Regression does not solve these identification problems. It draws on DiscoSCM's separation of an individual selection variable and event-level noise to define a structured predictor, then evaluates that predictor on robustness tasks.

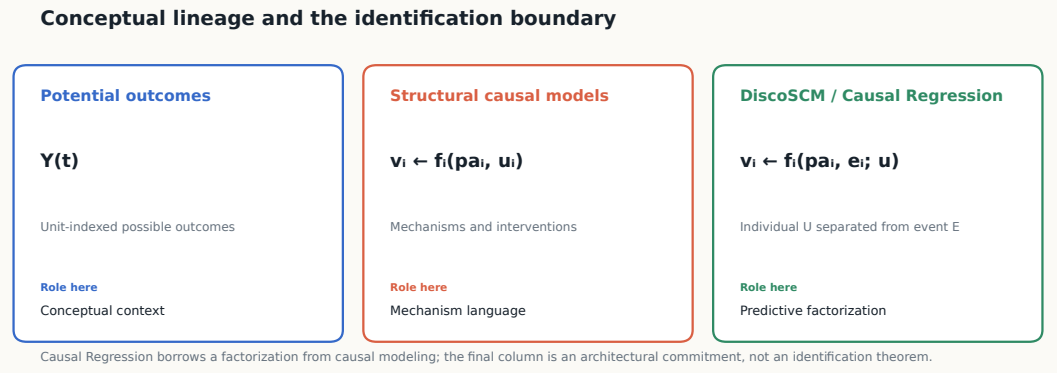

Figure 2: Conceptual lineage of Causal Regression. Potential Outcomes provide unit-indexed possible outcomes, SCMs provide mechanism language, and DiscoSCM motivates separating individual U from event noise E. The final column is an architectural commitment, not an identification result.

Robust regression and noisy labels. Classical robust losses and estimators reduce sensitivity to large residuals (Huber, 1964; Koenker & Bassett, 1978); tree ensembles offer a different inductive bias (Breiman, 2001; Friedman, 2001). Neural approaches to noisy labels often estimate sample reliability, regularize predictions, or model the corruption process. Our experiments compare CausalEngine with robust MLP losses, standard neural regressors, and tree ensembles. The matched-loss comparisons are especially important: they help separate a possible effect of the CausalEngine factorization from the effect of using a Cauchy objective alone.

2 The Causal Regression Framework

Statistical regression estimates predictive functionals such as $\mathbb{E}[Y \mid X]$. This is the appropriate target for many prediction problems, but it does not by itself distinguish stable mechanisms from environment-specific associations (Pearl, 2009). Invariant and causal prediction methods ask whether additional structure can improve stability under distribution change (Arjovsky et al., 2019; Peters et al., 2016; 2017).

Causal Regression introduces such structure through a latent individual variable U and an outcome mechanism:

$$Y = f(U, \varepsilon), \tag{2}$$

where ε represents residual event-level or environmental variation. This form is motivated by DiscoSCM (Gong et al., 2024), in which a stable individual selection variable is separated from exogenous noise.

For an identifiable query under a specified DiscoSCM, Population-Level Valuation follows an Abduction–Valuation–Reduction calculation:

1. **Abduction**: infer a distribution over U from observed evidence;
2. **Valuation**: evaluate the specified structural equation for a given U;
3. **Reduction**: marginalize the remaining uncertainty to obtain the target functional.

Our predictive architecture is inspired by this calculation. The theorem does not imply that a latent variable learned from arbitrary observational regression data is identified. That stronger interpretation would additionally require a correctly specified model, sufficient evidence, and the relevant identifiability assumptions.

We therefore distinguish the framework's *causal semantics* from what the present experiments establish. Within a specified model:

1. U denotes a latent individual representation intended to contain stable outcome-relevant information;
2. f denotes the parameterized mechanism mapping U and ε to the outcome;
3. ε denotes residual variation not represented by U.

In the empirical study, these are learned components of a predictive model. Calling U a causal representation states the model's intended semantics; it is not a claim that observational risk minimization uniquely recovers the true data-generating cause.

The learnable problem is decomposed into two maps:

1. an **Abduction model** $g : \mathcal{X} \to \mathcal{P}(\mathcal{U})$, which parameterizes a conditional distribution over U from features X;
2. a **Prediction mechanism** $f : \mathcal{U} \times \mathcal{E} \to \mathcal{Y}$, which maps the latent representation and residual variation to the outcome.

The two maps are trained jointly from predictive data. Without auxiliary causal supervision, interventions, multiple environments, or stronger structural restrictions, the optimized decomposition need not be unique.

CausalEngine is a concrete parameterization of g and f. Its Cauchy family and linear Action stage make the predictive distribution analytically tractable; the next section gives the exact construction.

3 The CausalEngine

CausalEngine turns Equation 2 into an end-to-end trainable regressor. The design addresses three questions: how to parameterize the conditional representation $P(U \mid X)$, how to map U to an outcome score, and how to propagate uncertainty without Monte Carlo sampling.

3.1 The dual role of U: selection and representation

In DiscoSCM, U indexes heterogeneous individuals separately from event-level noise (Gong et al., 2024). In CausalEngine, the same symbol also denotes a learned representation derived from X. This gives the architecture an intended individual-level semantics: variation in U represents uncertainty about stable individual attributes, whereas ε represents residual variation conditional on those attributes.

The distinction is meaningful only relative to the model assumptions. In particular, interpreting the two scales as epistemic and aleatoric uncertainty requires adequate specification of the representation and outcome mechanism, separation of U and ε, and no omitted source that is incorrectly absorbed into one component. We therefore evaluate these quantities as model components rather than claiming that their real-world semantics are identified from predictive data alone. Appendix C summarizes this boundary.

3.2 Core technical choices for tractable learning

3.2.1 Cauchy conditional representation

We use a Cauchy family because it combines heavy tails with stability under affine transformations. Heavy tails reduce the influence of large residuals, while affine stability permits analytical propagation through the linear Action stage. This is a technical modeling choice; we do not claim that the Cauchy family is a uniquely causal distribution.

3.2.2 Linear Action mechanism

The Action stage is linear in the learned representation. This is a restrictive but explicit approximation: an expressive Perception network is asked to learn a representation in which a scalar decision is

well described by a linear map. The choice is inspired by the practical success of linear heads over learned representations (Ouyang et al., 2022), but that analogy does not imply that all physical causal mechanisms are linear.

Suppose the components U_j are modeled as independent Cauchy variables with locations μ_j and scales γ_j, and the scalar score is

$$S = \sum_{j=1}^{d} w_j U_j + b. \tag{3}$$

Affine stability gives

$$S \sim \text{Cauchy}\left(\sum_{j=1}^{d} w_j \mu_j + b, \sum_{j=1}^{d} |w_j| \gamma_j \right). \tag{4}$$

The componentwise absolute value and sum in the scale are important; a single ambiguous expression such as $|W|\gamma$ is insufficient when U is a vector. Appendix B records the distributional assumptions behind this calculation.

3.3 CausalEngine four-stage architecture

3.3.1 Stage 1: Perception

Perception maps raw features to an expressive representation:

$$Z = \text{Perception}(X). \tag{5}$$

The module may be a multilayer perceptron or another encoder appropriate for the input. It must retain information useful for the downstream conditional model; the architecture alone does not guarantee that Z identifies a unique latent cause.

3.3.2 Stage 2: Abduction

Abduction parameterizes an approximate conditional distribution rather than solving an analytically known inverse structural equation:

$$q_\phi(U \mid Z) = \text{Cauchy}\big(\mu_U(Z), \gamma_U(Z)\big). \tag{6}$$

The location and positive scale are learned from Z:

$$\mu_U(Z) = W_{\text{loc}} Z + b_{\text{loc}}, \tag{7}$$

$$\gamma_U(Z) = \text{softplus}(W_{\text{scale}} Z + b_{\text{scale}}). \tag{8}$$

The nonlinear Perception module can encode complex feature dependence; the two heads parameterize the chosen Cauchy approximation. Thus the method does not claim an exact posterior for an otherwise unknown nonlinear inverse problem.

3.3.3 Stage 3: Action

The model adds componentwise Cauchy residual noise and applies the linear map:

$$U' = U + \varepsilon, \qquad \varepsilon_j \sim \text{Cauchy}(0, b_{\text{noise},j}), \tag{9}$$

$$S = W_{\text{action}} U' + b_{\text{action}}. \tag{10}$$

Under the componentwise independence assumptions, the score remains Cauchy. For a row vector $W_{\text{action}} = (w_1, \ldots, w_d)$, its parameters are

$$\mu_S = \sum_j w_j \mu_{U,j} + b_{\text{action}}, \tag{11}$$

$$\gamma_S = \sum_j |w_j| (\gamma_{U,j} + b_{\text{noise},j}). \tag{12}$$

This closed form avoids sampling variance in the propagation step.

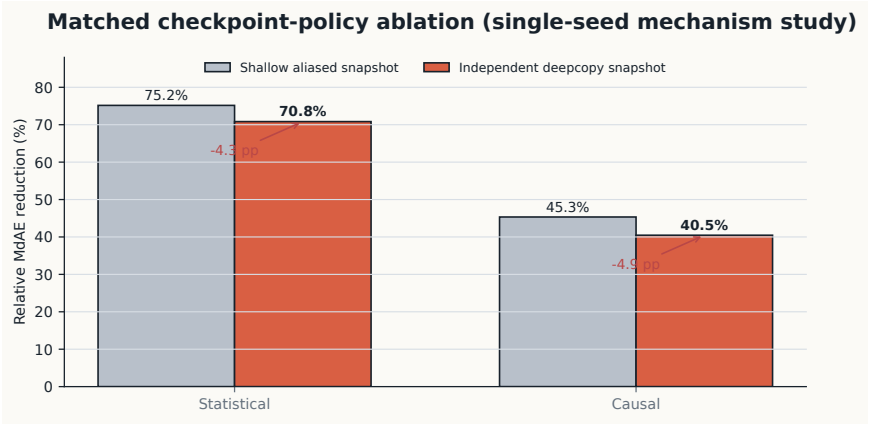

Figure 3: Matched checkpoint-policy ablation at 40% shuffle. The shallow tensor-aliased snapshot is compared with an independent deep-copied snapshot while data, seed, device, shuffle semantics, and other repaired-source settings remain fixed. This single-seed panel diagnoses the mechanism behind the result change; it is not the five-seed submission estimate.

3.3.4 Stage 4: Decision

Decision maps the score distribution to the task output. For regression we use the identity transformation $Y = S$ and minimize the Cauchy negative log-likelihood:

$$\mathcal{L}_{\text{regression}} = \log(\pi\gamma_S) + \log\left[1 + \left(\frac{y - \mu_S}{\gamma_S}\right)^2\right]. \tag{13}$$

Its logarithmic growth reduces the influence of large residuals, similarly to a robust M-estimator. Because robustness may arise from the loss as well as the factorization, the experiments include Cauchy-loss MLP and within-architecture loss comparisons where available.

4 Experiments

We re-evaluate the original Causal Regression experiments using the corrected implementation. The purpose is to preserve the original scientific questions while replacing historical point estimates that depended on ineffective best-checkpoint restoration.

4.1 Evaluation protocol and implementation correction

The historical implementation saved early-stopping parameters using `state_dict().copy()`. The copied dictionary retained references to live tensors, so subsequent optimization also changed the nominal best checkpoint. The corrected implementation uses a deep copy and restores the independent validation-best parameters. A controlled 624-condition Synthetic ablation isolates this checkpoint policy as the main direct source of the reduced historical advantage; restore-at-every-exit behavior has no measurable effect in that matrix because all neural fits stop through patience.

The paper-compatible shuffle corruption remains a global permutation of the training-label vector and may contain fixed points. This is the intended protocol, not an implementation defect. We treat fixed-point-free derangement as a separate sensitivity analysis rather than silently changing the paper condition.

Every corrected run records its source-tree hash, random seed, checkpoint policy, optimizer parameters, dataset provenance, package versions, and execution device. Statistical uncertainty is estimated over independent random seeds. Repeating one seed on several machines is used only to assess runtime portability and numerical drift.

For the main comparison, `CausalEngine (standard)` is fixed in advance as the primary method. We additionally report the best CausalEngine mode for each condition as a descriptive

Table 1: Corrected Synthetic MdAE at 40% training-label corruption over five paired seeds. Lower is better; positive advantage favors the fixed CausalEngine standard mode.

Generator	Corruption	Standard	Best baseline	Paired advantage [95% CI]	Wins	Family oracle
Causal	asymmetric	2.9039	2.0067	-0.8972 [-1.6656, -0.1287]	20%	2.1548
Causal	outlier	1.3154	0.6282	-0.6872 [-1.2409, -0.1335]	0%	1.0542
Causal	shuffle	0.2267	0.3837	0.1570 [0.0946, 0.2193]	100%	0.2150
Causal	systematic	3.0041	2.1880	-0.8161 [-1.5390, -0.0932]	20%	2.0244
Statistical	asymmetric	96.3341	79.0466	-17.2875 [-39.7830, 5.2080]	20%	64.9910
Statistical	outlier	30.8588	19.6590	-11.1998 [-26.9827, 4.5831]	0%	28.1559
Statistical	shuffle	3.3950	10.8609	7.4659 [6.9022, 8.0295]	100%	3.3950
Statistical	systematic	88.8205	68.2840	-20.5365 [-43.4802, 2.4072]	0%	60.7691

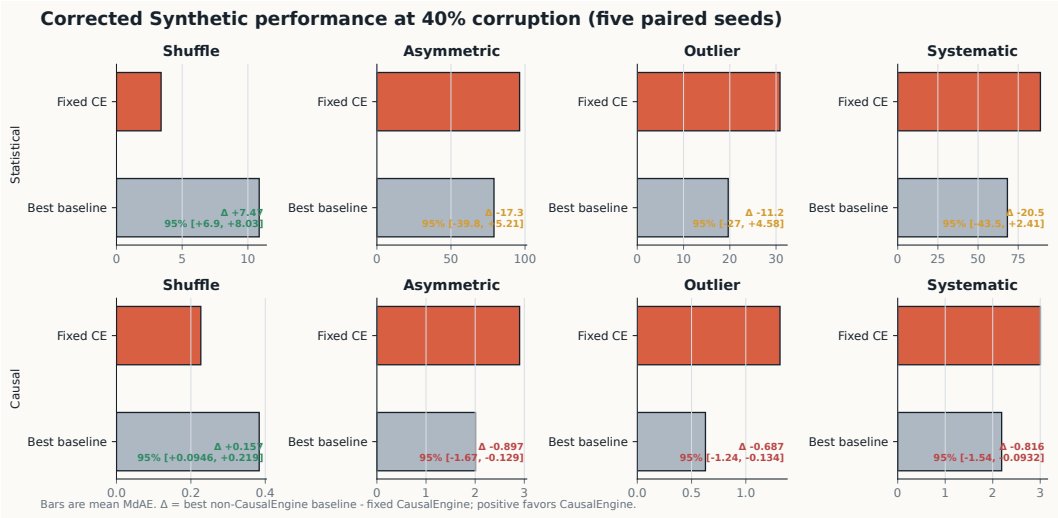

Figure 4: Corrected Synthetic MdAE at 40% corruption over five paired seeds. Each panel compares the pre-specified fixed standard CausalEngine with the strongest non-CausalEngine baseline; annotations give the paired advantage and its 95% interval. Only shuffle is consistently positive on both generators.

family-oracle comparison. This prevents mode selection from being confused with a pre-specified algorithm result. Our primary metric remains Median Absolute Error (MdAE); tables also include MAE, RMSE, and R^2 where space permits.

4.2 Synthetic data: corrected label-noise robustness

We retain the original Statistical and Causal synthetic generators, four label corruption families, baseline set, and noise-ratio grid. With independent best-checkpoint restoration, the shuffle result remains positive but is materially smaller than the historical report. At 40% shuffle, the fixed standard mode wins all five paired seeds on both generators. On Statistical data, its mean MdAE is 3.3950 versus 10.8609 for the strongest baseline; the mean per-seed relative reduction is 68.9% with 95% interval $[64.6\%, 73.2\%]$, replacing the historical 84.0%. On Causal data, the means are 0.2267 and 0.3837, and the relative reduction is 40.4% $[28.4\%, 52.3\%]$, replacing 55.7%.

This advantage does not extend uniformly to the other corruption families. For the Causal generator, the fixed standard interval is negative under asymmetric, outlier, and systematic corruption. For the Statistical generator, the corresponding means are also negative, but their intervals cross zero. The corrected Synthetic result is therefore shuffle-specific rather than a general label-noise robustness result.

Table 2: Corrected California Housing MdAE at 40% training-label corruption over paired seeds. Lower is better; positive advantage favors the fixed CausalEngine standard mode.

Condition	Standard	Best baseline	Paired advantage [95% CI]	Wins	Family oracle
asymmetric	0.6652	0.5467	-0.1185 [-0.2282, -0.0087]	0%	0.5278
outlier	0.4682	0.2587	-0.2095 [-0.4330, 0.0140]	0%	0.3451
shuffle	0.2270	0.2647	0.0377 [0.0207, 0.0547]	100%	0.2221
systematic	0.7048	0.5579	-0.1469 [-0.2424, -0.0513]	0%	0.5537

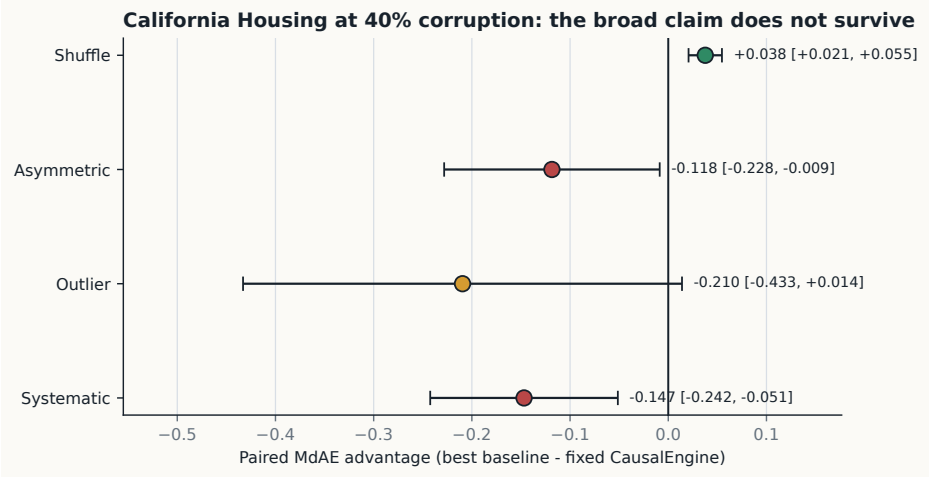

Figure 5: Corrected California Housing paired advantage at 40% corruption. Points show the five-seed mean and bars show 95% paired intervals. Positive values favor the fixed standard CausalEngine. The figure visually retracts the historical claim of consistent superiority across noise types.

4.3 California Housing: corruption-specific results

The corrected California Housing matrix does not reproduce the original claim of consistent superiority across diverse corruption types. Over five paired seeds, the fixed standard mode wins all five 40% shuffle comparisons against the strongest non-CausalEngine baseline, Pinball MLP. Its mean MdAE is 0.2270 versus 0.2647, a paired advantage of 0.0377 with 95% interval $[0.0207, 0.0547]$. The direction reverses under the other corruption types: the fixed standard mode wins none of the five seeds for asymmetric, outlier, or systematic noise. Its paired interval is strictly negative for asymmetric and systematic corruption; the outlier interval crosses zero, while the best-of-CausalEngine family oracle is also significantly worse. We therefore report the negative results and restrict the supported robustness claim to label permutation in this experiment.

4.4 Public datasets

We retain the eight public regression datasets and the 0%/40% shuffle conditions from the original paper, with pinned OpenML identifiers, versions, checksums, target columns, and dataset summaries. Over five paired seeds, the fixed standard mode has a strictly positive 95% interval on six datasets: California, Concrete, CPU Small, Diamonds, Kin8nm, and Tecator. It wins all five seeds on each of these datasets. On Bodyfat it wins none and is significantly worse. On Space GA it wins three of five and the interval crosses zero. The descriptive family oracle has a positive interval on seven of eight datasets, including Space GA, but this selection-favorable count is not a fixed-algorithm result. The historical "up to three-fold" statement is also removed; the corrected fold advantages are below two.

Table 3: Corrected public-dataset MdAE at 40% training-label permutation over paired seeds. Lower is better; positive advantage favors the fixed CausalEngine standard mode.

Condition	Standard	Best baseline	Paired advantage [95% CI]	Wins	Family oracle
bodyfat	3.6119	1.7357	-1.8763 [-3.0495, -0.7030]	0%	2.1813
california	0.2270	0.2647	0.0377 [0.0207, 0.0547]	100%	0.2221
concrete	3.9838	4.3818	0.3979 [0.0897, 0.7062]	100%	3.6854
cpu_small	1.5369	1.9129	0.3759 [0.0795, 0.6724]	100%	1.4905
diamonds	107.4605	173.9354	66.4749 [36.7344, 96.2155]	100%	107.4605
kin8nm	0.0553	0.0627	0.0073 [0.0020, 0.0127]	100%	0.0537
space_ga	0.0674	0.0694	0.0020 [-0.0055, 0.0095]	60%	0.0622
tecator	1.3844	2.1033	0.7188 [0.0655, 1.3721]	100%	1.3050

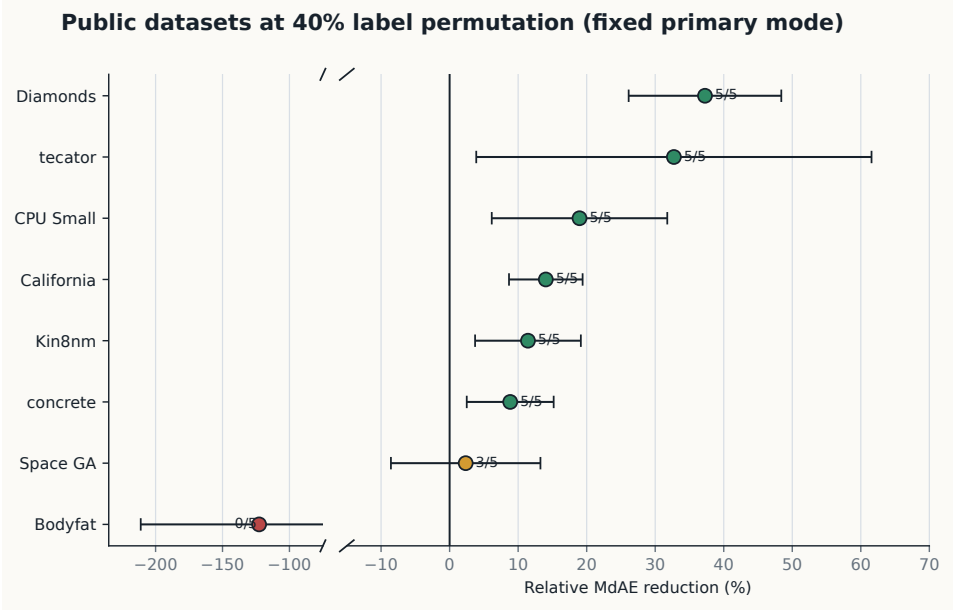

Figure 6: Five-seed relative MdAE reduction of the fixed standard CausalEngine on eight public datasets at 40% label permutation. Points are paired means, bars are 95% intervals, and labels give paired wins. The broken axis isolates Bodyfat so the six positive datasets remain readable.

4.5 Interpretation boundary

The corrected experiments test predictive degradation after constructed label corruption. They do not by themselves identify the learned representation as the true latent cause, recover a structural equation, or establish valid interventional or counterfactual predictions. The empirical conclusion is therefore about robustness of this architecture under the tested prediction tasks, not direct verification of causal-mechanism recovery.

5 Conclusion and Discussion

This paper introduced Causal Regression and CausalEngine, a structured regression architecture that separates a latent individual representation from residual environmental variation and uses a tractable Cauchy parameterization with a linear mechanism. The corrected empirical study shows why the strength of the conclusion must be narrower than in the historical submission.

An implementation audit found that shallow tensor-aliased checkpoints did not preserve an independent validation-best model. Correct restoration benefits several baselines more than CausalEngine and materially reduces the originally reported relative margins. Nevertheless, the corrected experiments

retain a robustness signal under training-label permutation. The same evidence does not support uniform superiority under asymmetric, outlier, and systematic corruption on California Housing.

The resulting conclusion is conditional. Across five paired seeds, the fixed standard mode is better under 40% label permutation on both Synthetic generators and has a strictly positive interval on six of eight public datasets. California Housing also shows that this is not a general label-corruption result: the model wins all five shuffle comparisons and none of the asymmetric, outlier, or systematic comparisons. These experiments do not establish universal robustness or direct recovery of causal mechanisms.

Future work should test whether the observed behavior persists under feature-dependent and subgroup-dependent corruption, expand the evaluation to additional datasets, and directly assess calibration, interventions, or counterfactual predictions before promoting stronger causal claims. It should also study which parts of the factorization, loss, and uncertainty parameterization drive the remaining label-permutation robustness.

REFERENCES

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Christopher M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.

Clark Glymour, Kun Zhang, and Peter Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in genetics*, 10:524, 2019.

Heyang Gong, Chaochao Lu, and Yu Zhang. Distribution-consistency structural causal models. *arXiv preprint arXiv:2401.15911*, 2024.

Trevor Hastie, Robert Tibshirani, Jerome Friedman, et al. The elements of statistical learning, 2009.

Peter J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35 (1):73–101, 1964.

Roger Koenker and Gilbert Bassett, Jr. Regression quantiles. *Econometrica*, 46(1):33–50, 1978.

Jerzy Neyman. On the application of probability theory to agricultural experiments. essay on principles. section 9. *Roczniki Nauk Rolniczych*, 10:1–51, 1923. English translation by D. M. Dabrowska and T. P. Speed in Statistical Science, 1990, 9, 465–480.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022.

Nick Pawlowski, Damien Causal, Ilyes Khemakhem, Martin Gütschow, Niki Kilbertus, Bernhard Schölkopf, and Peter Bühlmann. Deep structural causal models for tractable counterfactual inference. In *Advances in Neural Information Processing Systems*, volume 33, pp. 15135–15146, 2020.

Judea Pearl. *Causality*. Cambridge university press, 2009.

Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):947–1012, 2016.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

Audrey Poinsot, Alessandro Leite, Nicolas Chesneau, Michèle Sébag, and Marc Schoenauer. Learning structural causal models through deep generative models: Methods, guarantees, and challenges. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24)*, pp. 8207–8215, 2024. doi: 10.24963/ijcai.2024/907.

Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688, 1974.

Peter Spirtes, Clark N Glymour, and Richard Scheines. *Causation, prediction, and search*. MIT press, 2000.

Kevin Xia, Kai-Zhan Lee, Yoshua Bengio, and Elias Bareinboim. The causal-neural connection: Expressiveness, learnability, and inference. *Advances in Neural Information Processing Systems*, 34:25389–25401, 2021.

Jie Zhong, Weihua Shen, Yaliang Li, and Dacheng Tao. A comprehensive survey of instruction-following language models: Architectures, applications, and future trends. *arXiv preprint arXiv:2504.12328*, 2025.

A Corrected implementation and evidence ledger

The new submission keeps the historical source and the corrected implementation as separate evidence chains. Historical artifacts describe how the submitted numbers were produced; corrected artifacts determine the claims of this paper. The corrected chain stores an independent deep-copied best checkpoint, restores it at termination, applies the configured random state to Torch, NumPy, data splits, corruptions, and tree baselines, and records data provenance and device metadata in each run manifest.

The paper-compatible shuffle protocol permutes the complete training-label vector using a private seeded random-number generator. Selected labels may remain at their original indices through permutation fixed points. A fixed-point-free subset derangement is retained only as a separately named sensitivity protocol.

A.1 Result promotion gate

A numerical claim is promoted into the final paper only when the corresponding matrix is complete for every planned seed, all runs share one corrected source hash and protocol, public-dataset provenance matches exactly, and paired uncertainty is reported. Cross-host repeats of a single seed are not counted as independent observations. Best-of-CausalEngine results are labelled as family oracle comparisons and never substituted for the fixed primary model.

A.2 Promoted result tables

All fifteen planned manifests—five seeds for Synthetic, California, and the public-dataset matrix—passed the promotion gate with one corrected source hash. The generated tables in Section 4 are byte-identical to the aggregator outputs. Historical high-noise tables and curves are not copied into this corrected appendix because they reflect the aliased-checkpoint training behavior.

A.3 Corrected robustness curves

B Properties of the Cauchy distribution

For location $\mu \in \mathbb{R}$ and scale $\gamma > 0$, the Cauchy density is

$$p(x; \mu, \gamma) = \frac{1}{\pi\gamma \left[1 + \left(\frac{x-\mu}{\gamma}\right)^2\right]}. \tag{14}$$

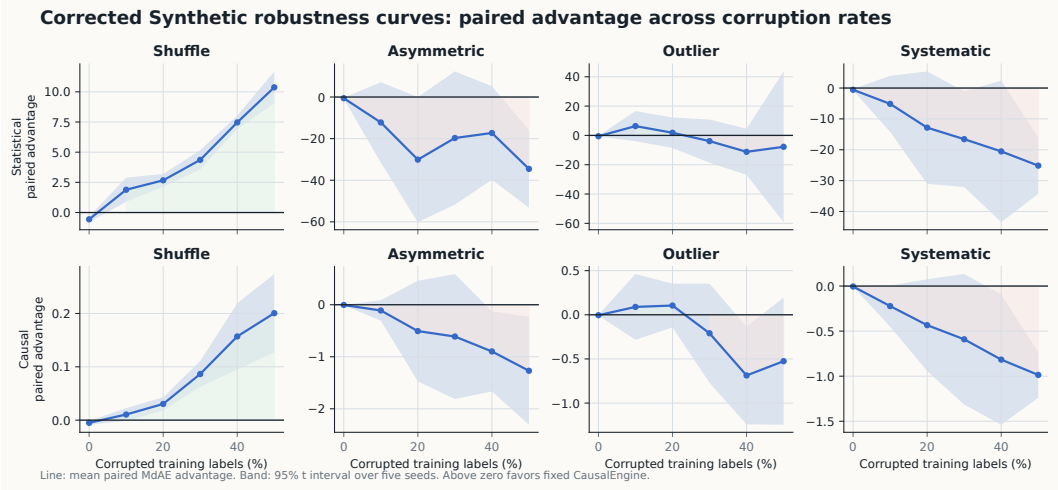

Figure 7: Corrected Synthetic paired MdAE advantage over the full corruption grid. Lines are five-seed means and bands are 95% paired intervals. Values above zero favor the fixed standard CausalEngine. The eight panels replace the historical single-seed robustness curves.

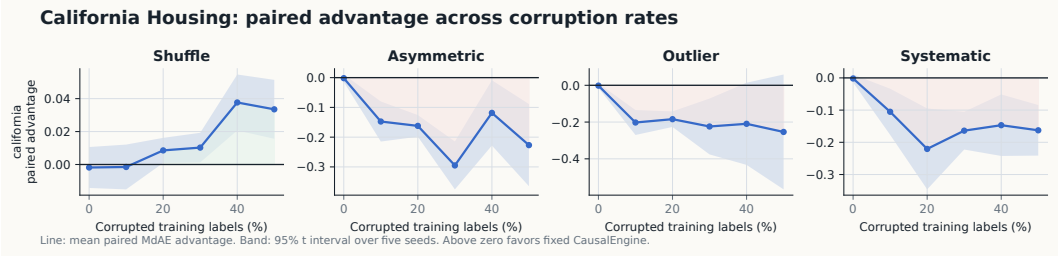

Figure 8: Corrected California Housing paired MdAE advantage across corruption rates. The positive shuffle trajectory and negative trajectories for the other corruption families make the condition-specific conclusion explicit.

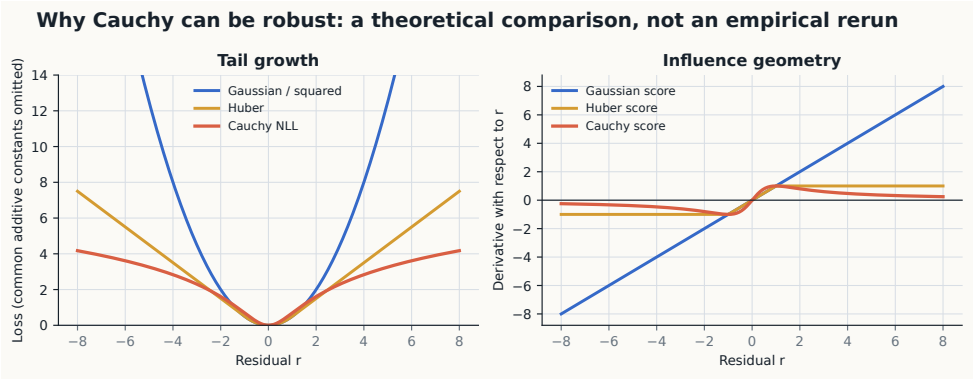

Figure 9: Theoretical loss and influence geometry for Gaussian/squared, Huber, and Cauchy objectives. The Cauchy score decreases for sufficiently large residuals, illustrating a possible source of robustness. This is an analytical comparison, not a corrected empirical Cauchy-versus-Normal rerun.

Its tails decay polynomially, and its mean and variance are undefined. The location μ is therefore not an expectation; it is also the median and mode of the symmetric distribution. In the experiments, point predictions use the predicted location rather than a nonexistent Cauchy mean.

The tractability of CausalEngine follows from stability under linear combinations. Let $X_j \sim \text{Cauchy}(\mu_j, \gamma_j)$ be mutually independent. For real coefficients a_j,

$$\sum_{j=1}^{d} a_j X_j \sim \text{Cauchy}\left(\sum_{j=1}^{d} a_j \mu_j, \sum_{j=1}^{d} |a_j| \gamma_j\right). \tag{15}$$

The absolute value is componentwise and the scales are summed. This identity requires the joint assumptions used by the implementation; marginally Cauchy components with arbitrary dependence do not in general justify the same scale formula.

For the Action model

$$U'_j = U_j + \varepsilon_j, \tag{16}$$

$$S = \sum_j w_j U'_j + b, \tag{17}$$

with independent $U_j \sim \text{Cauchy}(\mu_{U,j}, \gamma_{U,j})$ and $\varepsilon_j \sim \text{Cauchy}(0, b_{\text{noise},j})$, repeated application of Equation 15 gives

$$\mu_S = \sum_j w_j \mu_{U,j} + b, \tag{18}$$

$$\gamma_S = \sum_j |w_j| (\gamma_{U,j} + b_{\text{noise},j}). \tag{19}$$

This analytical propagation removes the need to sample U or ε in the predictive forward pass. It is a property of the chosen distribution and linear mechanism, not evidence that the learned representation is causally identified.

C Causal Modeling Frameworks

Potential Outcomes (PO) (Neyman, 1923; Rubin, 1974) and Structural Causal Models (SCMs) (Pearl, 2009) provide two standard languages for causal queries. PO indexes a unit's possible outcomes by a treatment value. If T is the observed treatment, the consistency assumption relates the observed and potential outcomes:

Assumption 1 (Consistency). $T = t \implies Y(t) = Y$.

An SCM is a tuple $\langle \mathbf{U}, \mathbf{V}, \mathcal{F}, P(\mathbf{U}) \rangle$. The endogenous variables $\mathbf{V}$ are assigned by structural equations

$$v_i \leftarrow f_i(pa_i, u_i), \tag{20}$$

where pa_i denotes endogenous parents and u_i denotes background variables. An intervention $do(X = x)$ replaces the structural equation for X by the constant assignment $X \leftarrow x$, leaving the other mechanisms unchanged. The resulting submodel defines potential outcomes such as $Y(x)$.

For a fully specified SCM, a joint counterfactual distribution can be valued by summing the background-variable distribution over the compatible states:

$$P(\mathbf{y}_{\mathbf{x}}, \ldots, \mathbf{z}_{\mathbf{w}}) = \sum_{\{\mathbf{u}:\mathbf{Y}(\mathbf{x})=\mathbf{y},\ldots,\mathbf{Z}(\mathbf{w})=\mathbf{z}\}} P(\mathbf{u}). \tag{21}$$

This valuation presumes the relevant mechanisms and background distribution are specified. Observational data do not automatically identify them, especially for joint counterfactual queries.

DiscoSCM (Gong et al., 2024) separates a unit selection variable U from event-level exogenous variables $\mathbf{E}$:

$$v_i \leftarrow f_i(pa_i, e_i; u). \tag{22}$$

Each $U = u$ denotes an individual, whereas repeated events for the same individual may vary through E. The framework uses distribution-consistency:

Assumption 2 (Distribution-consistency (Gong et al., 2024)). $X = x, U = u \implies Y(x) \stackrel{d}{=} Y$.

Unlike ordinary consistency, this is conditional on the unit and equates distributions rather than realized values.

Under a specified DiscoSCM, population-level valuation decomposes into individual valuation and reduction:

Theorem 1 (Population-Level Valuation (Gong et al., 2024)). *Let e denote factual evidence. If $P(u \mid e)$ and the individual response distribution $P(Y(x) = y; u)$ are defined by the model, then*

$$P(Y(x) = y \mid e) = \int P(Y(x) = y; u)\, dP(u \mid e). \tag{23}$$

This is the Abduction–Valuation–Reduction structure that motivates Causal Regression. In CausalEngine, $q_\phi(U \mid X)$ and the response kernel are learned parameterizations. Equation 23 explains how to reduce over U once those components have valid semantics; it does not prove that predictive risk minimization identifies them. Identification or external validation remains an additional requirement for interventional or counterfactual use.

D Language and tool support

Language models were used as auxiliary tools for editing, code assistance, and literature triage. The authors remain responsible for the research questions, implementation choices, experiment design, source verification, numerical analysis, and claims. In particular, all reported tables are generated from the archived run manifests and CSV outputs rather than from language-model estimates.