Robust Deep Neural Network Estimation under Heavy-Tailed Noise: Breakdown Point Analysis, Theory, and Empirical Evidence

Abdullah Mohammed Rashid

Department of Accounting and Financial Control, College of Business Economics, Al-Nahrain University, Iraq.

[1]*Received: 21/08/2026; Accepted: 29/09/2026; Published: 23/10/2026*

Abstract

Deep neural network (DNN) regressors are routinely trained with the squared-error loss, yet many applied targets (prices, returns, claims, clinical measurements) carry heavy-tailed noise and gross recording errors. This paper studies the robustness of DNN regression estimators from three complementary angles. Theoretically, we show that any stationary point of the empirical squared-error risk of a network with an output bias has finite-sample breakdown point 1/n under response contamination, whereas gradient-trained networks with a bounded score function ψ remain bounded under arbitrary label contamination, and become exactly invariant to the magnitude of outliers once their residuals saturate ψ. Because these properties make the classical breakdown point uninformative for practical networks, we adopt quantitative criteria: an empirical breakdown point, the sensitivity curve and Friedman–Nemenyi rank comparisons. Empirically, we train a 128-64-32 ReLU network with eight losses (squared error, Huber, adaptive Huber, least absolute deviations, log-cosh, Cauchy, Tukey bisquare and an LTS-type trimmed loss) on California Housing with 0–50% contaminated labels under three contamination mechanisms, with five independent seeds, and complement this with a Monte Carlo study of seven heavy-tailed noise laws, an ablation of common training choices, and seven classical and ensemble benchmarks. The squared-error network breaks down at the first contamination level (5%) in all five seeds, with a 50-fold increase in test error. Under symmetric Cauchy contamination all bounded-ψ losses remain stable up to 50%, but under one-sided or clustered contamination the monotone losses (Huber, log-cosh, LAD) break down at 20–40%, whereas the redescending Tukey and Cauchy losses and trimmed training do not break down at any tested level up to 40%. Tukey bisquare attains the best mean rank (2.14 of 8), significantly ahead of the Huber loss (5.38; Nemenyi CD = 1.48). Early stopping on a robustly scored validation split reduces the squared-error damage by a factor of 21–36 but does not prevent breakdown. We recommend redescending or trimmed objectives, with a robust warm start, as the default for DNN regression on data that may contain gross errors.

Keywords: *breakdown point; robust regression; M-estimation; Huber loss; Tukey bisquare; deep neural networks; heavy-tailed noise; label contamination; influence function.*

[1] *How to cite the article:* Rashid A.M; October 2026; Robust Deep Neural Network Estimation under Heavy-Tailed Noise: Breakdown Point Analysis, Theory, and Empirical Evidence; *International Journal of Innovations in Scientific Engineering*, Jul-Dec 2026, Vol 24, 41-64

1. Introduction

Many applied deep learning regression problems involve training labels that cannot be fully trusted: recording and transcription errors, sensor faults, censored or top-coded values, and data poisoning all generate observations whose responses bear little relation to the regression function. Such problems are common in real estate valuation, financial forecasting and clinical prediction, where the noise itself is often heavy-tailed. Robust statistics offers two classical tools for describing how an estimator reacts to such data. The influence function, introduced by Hampel [1] building on Huber's minimax theory of M-estimation [2], describes the effect of an infinitesimal contamination at a point, while the breakdown point, in its finite-sample replacement form due to Donoho and Huber [3], is the smallest fraction of observations that can drive the estimator arbitrarily far from its uncontaminated value [4, 5]. These tools have been studied in depth for linear and nonparametric regression, but they are rarely applied to over-parameterised DNN estimators trained by stochastic gradient methods [6].

Training with the squared-error (MSE) loss remains the default in deep regression because it is smooth, it targets the conditional mean, and its statistical theory for DNNs is well developed [7]. Its weakness is equally classical: the score function of the squared loss, $\psi(r) = 2r$, is unbounded, so a single gross error can dominate the gradient. Robust losses with bounded or redescending score functions remove this weakness, and a growing body of theory establishes convergence of robust DNN estimators when the noise has only a few finite moments [8, 9]. What is still missing is a careful link between this theory and the breakdown behaviour that practitioners observe when networks are trained on contaminated data.

An earlier version of this study compared only the MSE and Huber losses, used a single training run per contamination level and a single (symmetric) contamination mechanism, and stated breakdown values for the Huber loss that do not hold for regression with an unknown function. The present paper replaces that analysis with a broader and statistically controlled study. Its contributions are as follows.

1. **Theory.** We prove that every stationary point of the empirical MSE risk of a network with an output bias has finite-sample breakdown point 1/n under response contamination (Proposition 1); that mini-batch gradient training with any bounded score function yields networks that stay bounded under arbitrary label contamination of any fraction (Proposition 2); and that for monotone bounded and redescending losses the trained network becomes exactly invariant to the magnitude of outliers once their residuals reach the saturation region (Proposition 3). Together these results explain why the classical breakdown point does not discriminate between robust losses for DNNs, and motivate the quantitative criteria used in the rest of the paper.

2. **A wider family of estimators.** We compare eight losses: MSE, Huber with a fixed threshold, adaptive Huber with a data-driven threshold, least absolute deviations (LAD), log-cosh, Cauchy, Tukey bisquare, and an LTS-type trimmed loss, using the 95%-efficiency tuning constants of classical robust regression and a robust warm start for scale-dependent losses.

3. **Statistically controlled experiments.** Every condition is replicated over independent seeds, and the losses are compared with the Friedman test, Nemenyi critical differences and Holm-adjusted Wilcoxon signed-rank tests.

4. **Contamination mechanisms beyond symmetric noise.** In addition to symmetric Cauchy contamination, we study one-sided gross errors and clustered (point-mass) contamination, which separate monotone from redescending losses.

5. **A Monte Carlo study with a known regression function**, covering seven noise laws from Gaussian to Cauchy, so that the estimation error $\|\hat{f} - f^*\|$ can be measured directly.

6. **Sensitivity curves, training-choice ablations and benchmarks** against linear robust M-estimators, random forests and gradient boosting.

The remainder of the paper is organised as follows. Section 2 reviews related work. Section 3 develops the theoretical framework. Section 4 describes the experimental design. Section 5 reports and discusses the results, Section 6 discusses implications and limitations, and Section 7 concludes.

2. Related Work

Classical robust regression. The theory of M-estimation [2], the influence function [1, 4] and the breakdown point [3] were developed largely for location and linear regression models. For linear regression it is well known that least squares has breakdown point 1/n, that monotone M-estimators such as Huber and LAD regression resist outliers in the response ("vertical outliers") but still have breakdown point 1/n when leverage points are allowed, and that high-breakdown estimators such as least median of squares and least trimmed squares (LTS) [5, 10] and MM-estimators [11] attain breakdown points close to 1/2. Redescending score functions, such as Tukey's bisquare [12], combine high efficiency with strong resistance to gross errors, provided that a robust starting value and scale are available [13]. These lessons motivate the choice of losses and the warm-start procedure used here.

Robustness theory for DNN regression. Schmidt-Hieber [7] established near-minimax convergence rates for sparse ReLU networks under sub-Gaussian noise. Shen et al. [8] extended this to robust losses (including Huber, LAD and quantile losses) and derived non-asymptotic error bounds requiring only a finite p-th moment of the response. Fan, Gu and Zhou [9] quantified how the tail index of the noise affects the attainable convergence rate of DNN estimators and showed that an adaptive Huber loss, whose robustification parameter grows with the sample size, attains improved rates. Related results include error bounds for DNN regression on approximate manifolds [14], robust nonparametric regression with deep ReLU networks [15], Huber regression under weak conditional moments [16], functional linear Huber regression [17], and the theory of regression under Huber contamination in online settings [18]. Feng and Wu [19] studied robust nonparametric regression with heavy-tailed noise for a broad class of losses, including Tukey, Welsch and Barron losses [20], and showed that robustification is necessary when the hypothesis class contains unbounded functions. These works concern convergence rates rather than breakdown, and do not report empirical breakdown behaviour.

Breakdown and robust losses for neural networks. Werner [21] defined a regression breakdown point for feed-forward networks, computed it for several architectures and contamination models, and showed by simulation that robust losses improve out-of-sample accuracy under contamination. Rubio and Dorronsoro [22] compared robust losses for deep regression, and Wang, Zhou and Lin [23] proposed learning a data-adaptive loss for deep regression. Tyralis et al. [24] introduced deep Huber quantile regression networks. Werner's study did not consider heavy-tailed noise laws such as the Cauchy distribution, asymmetric contamination or replicated experiments on a real benchmark dataset, and none of the above compares monotone, redescending and trimmed objectives within one statistically controlled design.

Learning with noisy labels. The literature on learning with label noise [6, 25] is dominated by classification with symmetric or asymmetric label flipping. Two of its findings are directly relevant here: over-parameterised networks can memorise arbitrary labels [26], and gradient descent with early stopping is provably robust to a fraction of corrupted labels in over-parameterised networks [27]. Section 5.7 tests whether these implicit mechanisms protect MSE-trained regression networks against heavy-tailed label noise.

3. Theoretical Framework

3.1 Model, estimators and contamination

Let the data be generated by

$$Y_i = f^*(X_i) + \varepsilon_i, \qquad i = 1, \dots, n, \qquad (1)$$

where $X_i \in \mathcal{X} \subset \mathbb{R}^d$ is a bounded covariate vector, f^* is the unknown regression function, and the ε_i are i.i.d. errors that may be heavy-tailed. We say that ε has tail index α if $P(|\varepsilon| > t) = t^{-\alpha}L(t)$ for a slowly varying function L. Gaussian errors correspond to $\alpha = \infty$, Student-t_ν errors to $\alpha = \nu$, symmetric α-stable errors with $\alpha < 2$ to tail index α, and the Cauchy distribution to $\alpha = 1$, for which neither the mean nor the variance exists. Only moments of order below α are finite.

We consider ReLU multilayer perceptrons

$$f(x;\theta) = W_{L+1}\,\sigma(W_L \cdots \sigma(W_1 x + b_1) \cdots + b_L) + b_{L+1}, \qquad (2)$$

with $\sigma(u) = \max(u, 0)$, parameters $\theta = (W_1, b_1, \dots, W_{L+1}, b_{L+1})$ and, importantly, a scalar output bias b_{L+1}. Given a loss ρ with score function $\psi = \rho'$, the M-estimator is

$$\hat{f}_n = f(\cdot\,;\hat{\theta}), \qquad \hat{\theta} \in \operatorname*{argmin}_{\theta} \frac{1}{n}\sum_{i=1}^{n} \rho\left(\frac{f(X_i;\theta) - Y_i}{s}\right), \qquad (3)$$

where $s > 0$ is a residual scale (set to 1 for MSE, the fixed-threshold Huber loss, LAD and log-cosh, and estimated robustly for the other losses, see Table 1). In practice $\hat{\theta}$ is not a global minimiser but the output of a finite number of (mini-batch) gradient steps, a distinction that matters for breakdown, as shown below.

Contamination model. Let $\mathcal{Z}_n = \{(X_i, Y_i)\}_{i=1}^n$ be the sample and let $\mathcal{Z}_n^m$ denote any sample obtained by replacing m of the responses Y_i by arbitrary values, keeping the covariates fixed. This is response (label) contamination, the setting of all experiments below. Contamination of the covariates (leverage points) is a different problem and is not studied here.

Table 1. *Losses compared.* r is the residual divided by the scale s; $\hat{\sigma}$ = 1.4826 × MAD of the current training residuals, refreshed every epoch. Tuning constants give 95% asymptotic efficiency at the Gaussian model for the location problem.

Loss	$\rho(r)$	$\psi(r)$	**Bounded** ψ	**Redescending**	**Scale / tuning**
MSE	r^2	$2r$	No	No	none
Huber	$r^2/2$ if $\lvert r\rvert \le \delta$; $\delta(\lvert r\rvert - \delta/2)$ otherwise	$\max\big(-\delta, \min(r, \delta)\big)$	Yes	No	$\delta = 1$ (target units)
Adaptive Huber	as Huber	as Huber	Yes	No	$\delta = 1.345\,\hat{\sigma}$

Loss	$\rho(r)$	$\psi(r)$	**Bounded** ψ	**Redescending**	**Scale / tuning**
LAD (L1)	$\lvert r\rvert$	$\operatorname{sign}(r)$	Yes	No	none
Log-cosh	$\log\cosh(r)$	$\tanh(r)$	Yes	No	none
Cauchy	$c^2/2\log(1+r^2/c^2)$	$r/(1+r^2/c^2)$	Yes	Yes (slowly)	$c = 2.385\,\hat{\sigma}$
Tukey bisquare	$c^2/6\{1-(1-r^2/c^2)^3\}$ if $\lvert r\rvert \le c$; $c^2/6$ otherwise	$r(1-r^2/c^2)^2\,\mathbb{1}\{\lvert r\rvert \le c\}$	Yes	Yes (to 0)	$c = 4.685\,\hat{\sigma}$
Trimmed (LTS-type)	r^2 for the h smallest $\lvert r\rvert$ in each mini-batch, 0 otherwise	$2r$ or 0	—	Hard rejection	$h = \lfloor 0.5\,B \rfloor$

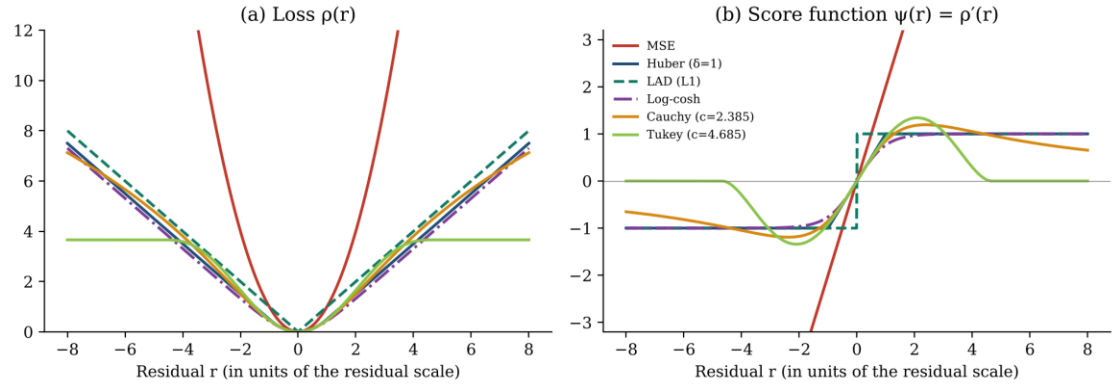


Figure 1. *Loss functions (a) and score functions (b) of the six smooth losses in Table 1. Monotone bounded scores (Huber, LAD, log-cosh) cap the influence of an outlier; redescending scores (Cauchy, Tukey) progressively remove it.*

3.2 Breakdown point and influence function

Following Donoho and Huber [3], the finite-sample replacement breakdown point of a function-valued estimator T under response contamination is

$$\varepsilon_n^*(T, \mathcal{Z}_n) = \frac{1}{n}\min\left\{m : \sup_{\mathcal{Z}_n^m} \sup_{x\in\mathcal{X}} |T(\mathcal{Z}_n^m)(x) - T(\mathcal{Z}_n)(x)| = \infty\right\}, \qquad (4)$$

with $\varepsilon_n^* = 1$ if no such m exists. For a parametric M-estimator with parameter θ, the influence function [1, 4] at a point (x, y) is

$$\mathrm{IF}\big((x,y); \theta, P\big) = -H^{-1}\,\psi(f(x;\theta) - y)\,\nabla_\theta f(x;\theta), \qquad H = \mathbb{E}_P[\nabla_\theta^2\,\rho(f(X;\theta) - Y)], \qquad (5)$$

whenever H is invertible. Equation (5) shows that the effect of a single observation is proportional to ψ evaluated at its residual. It is unbounded in y for MSE, bounded for monotone bounded scores, and zero for residuals beyond c under the Tukey loss. Note that boundedness of the influence function with respect to the response does not by itself

imply a positive breakdown point in regression; for linear M-estimators with monotone ψ, the breakdown point is still $1/n$ once leverage points are allowed [5, 13].

3.3 Main results

Proposition 1 (MSE stationary points break down with one observation). Let $\mathcal{F}$ be any network class of the form (2) with an output bias, and let $\hat{\theta}$ be any stationary point of the empirical MSE risk $\hat{R}(\theta) = n^{-1}\sum_i \big(Y_i - f(X_i;\theta)\big)^2$ at which $\hat{R}$ is differentiable with respect to b_{L+1}. Then

$$\frac{1}{n}\sum_{i=1}^{n} \hat{f}_n(X_i) = \bar{Y}_n. \qquad (6)$$

Consequently, if a single response Y_1 is replaced by $Y_1 + M$, every stationary point satisfies $\max_i|\hat{f}_n(X_i)| \geq |\bar{Y}_n + M/n| \to \infty$ as $M \to \infty$, so that $\varepsilon_n^* = 1/n$.

Proof. Since $\partial f(x;\theta)/\,\partial b_{L+1} = 1$ for every x, stationarity in b_{L+1} gives $0 = \partial\hat{R}/\,\partial b_{L+1} = -2/n\sum_i \big(Y_i - \hat{f}_n(X_i)\big)$, which is (6). Replacing Y_1 by $Y_1 + M$ adds M/n to the right-hand side of (6). The average of the fitted values, and therefore their maximum in absolute value, diverges with M, and the change in $\hat{f}_n$ over $\mathcal{X} \supseteq \{X_i\}$ is unbounded. ■

Proposition 2 (bounded scores give bounded networks under gradient training). Let ρ have a bounded score, $\sup_r|\psi(r)| \leq K$, and let θ_T be obtained from a fixed initialisation θ_0 by T steps of mini-batch (sub)gradient descent,

$$\theta_{t+1} = \theta_t - \frac{\eta_t}{|B_t|}\sum_{i\in B_t} \psi\big(f(X_i;\theta_t) - Y_i\big)\,\nabla_\theta f(X_i;\theta_t), \qquad (7)$$

with fixed step sizes η_t and a fixed sequence of mini-batches B_t. Assume $\| x \| \leq c_X$ on $\mathcal{X}$. Then there are constants R_T and H_T, depending on θ_0, (η_t), T, K, c_X and the architecture but not on the responses, such that $\| \theta_T \| \leq R_T$ and $\sup_{x\in\mathcal{X}}|f(x;\theta_T)| \leq H_T$ for every response vector. Hence the estimator has $\varepsilon_n^* = 1$ in the sense of (4): no fraction of contaminated labels can make it diverge.

Proof. For a ReLU network, every element of the (Clarke) subgradient $\nabla_\theta f(x;\theta)$ is a sum of products of weight matrices and inputs, so $G(R) := \sup\{\| \nabla_\theta f(x;\theta) \| : \| \theta \| \leq R, \| x \| \leq c_X\}$ is finite and non-decreasing in R. From (7), $\| \theta_{t+1} \| \leq \| \theta_t \| + \eta_t K\, G(\| \theta_t \|)$. Define $R_0 = \| \theta_0 \|$ and $R_{t+1} = R_t + \eta_t K G(R_t)$. By induction $\| \theta_t \| \leq R_t$ for all t, and R_T does not involve the responses. Finally, $|f(x;\theta)|$ is bounded on $\{\| \theta \| \leq R_T, \| x \| \leq c_X\}$ by a constant H_T. ■

For MSE the same recursion fails: the step contains the factor $2(f(X_i;\theta_t) - Y_i)$, so a single contaminated response in the first mini-batch moves the output bias by an amount proportional to the outlier, and $\varepsilon_n^* = 1/n$ also for the algorithmic estimator.

Remark (adaptive optimisers). Adam [28] normalises each coordinate of the update, so its per-step change is bounded by approximately the learning rate for any loss, including MSE. For a finite training budget, Adam-trained MSE networks are therefore also bounded in the strict sense of (4). The bound is loose, however: it grows with the number of steps and with depth, and Section 5.4 shows that an MSE network trained with Adam moves its predictions by more than 470 units (×\$100k) in response to one corrupted label out of 2,000. Breakdown in the strict sense (4) is thus too coarse a criterion for practically trained networks, and quantitative measures are needed.

Proposition 3 (saturation and exact invariance). Suppose $\psi(r) = K\,\mathrm{sign}(r)$ for $|r| \geq \delta$ (Huber, LAD, and log-cosh approximately), or $\psi(r) = 0$ for $|r| \geq c$ (Tukey bisquare, hard trimming). Let I be the set of contaminated observations and suppose that along the whole training trajectory every $i \in I$ has a residual in the saturation region with a fixed sign. Then the trajectory $(\theta_t)_{t \leq T}$, and hence $\hat{f}_n$, is identical for every choice of the contaminated responses that preserves this condition. In particular, shifting a corrupted label from $Y_i + M$ to $Y_i + M'$ with $M' > M$ large leaves the trained network unchanged.

Proof. Under the stated condition each contaminated observation contributes the term $-K\nabla_\theta f(X_i; \theta_t)$ (or 0) to (7), which does not depend on the value of Y_i. The recursion (7) is therefore the same function of θ_t for all such responses, and the trajectories coincide. ■

Proposition 3 identifies the residual effect of gross errors for monotone bounded losses: the outlier still pushes the fit with a constant force K in its direction. When the outliers have mixed signs and comparable magnitudes, as with symmetric Cauchy contamination, these pushes largely cancel. When they share a sign, as with one-sided or clustered contamination, they accumulate in proportion to the contamination fraction and bias the fit. Redescending and trimmed losses set the push to zero. This yields a testable prediction, which Section 5.5 confirms: monotone bounded losses should be robust to symmetric contamination but break down under asymmetric contamination at moderate fractions, while redescending and trimmed losses should not.

Relation to convergence rates. Robust losses also matter for the rate of convergence. For Lipschitz losses such as Huber and LAD, Shen et al. [8] obtained non-asymptotic error bounds for DNN estimators under only a finite p-th moment of the response, and Fan, Gu and Zhou [9] showed that the noise tail index determines the attainable rate and that an adaptive Huber loss whose threshold grows with n improves it. The Monte Carlo study in Section 5.6 measures this efficiency side of robustness directly.

3.4 Quantitative robustness criteria used in this paper

Given Propositions 2 and 3, we compare estimators with the following criteria, all computed on a clean, never-contaminated test set.

- **Empirical breakdown point.** The smallest tested contamination fraction at which the mean test MAE exceeds 1.5 times the same loss's mean MAE on clean training data.
- **Sensitivity curve.** The mean absolute change in test predictions, $\frac{1}{n_{te}}\sum_j \left|\hat{f}_M(x_j) - \hat{f}_0(x_j)\right|$, when m training labels are shifted by $+M$, plotted against M. This is the finite-sample analogue of (5) for $m = 1$.
- **Estimation error.** In simulations where f^* is known, the root-mean-square distance $\| \hat{f}_n - f^* \|$ on 5,000 noise-free test points.
- **Rank-based comparison.** Mean Friedman ranks over all (contamination level × seed) blocks, the Nemenyi critical difference, and Holm-adjusted Wilcoxon signed-rank tests against the Huber loss [29, 30].

4. Experimental Design

4.1 Dataset and preprocessing

The real-data benchmark is the California Housing dataset [31], derived from the 1990 U.S. census: 20,640 block groups, eight covariates (median income, median house age, average rooms, average bedrooms, population, average

occupancy, latitude and longitude), and the median house value in units of \$100,000 as the response (range 0.15–5.00). The design matrix was rebuilt from the raw census file so that it matches the variables of sklearn.datasets.fetch_california_housing() [32]. In the raw file, 207 block groups lack the bedroom count, and these average-bedroom values were imputed with the median. The data are split 80/20 into 16,512 training and 4,128 test observations with random_state = 42. Covariates are standardised with a scaler fitted on the training split only; the response is not transformed. The test set is never contaminated.

4.2 Network and training

All DNN estimators share the architecture of Figure 2: three fully connected hidden layers of 128, 64 and 32 ReLU units and a linear output unit, for a total of 11,521 parameters ($8\cdot128 + 128 + 128\cdot64 + 64 + 64\cdot32 + 32 + 32\cdot1 + 1$). Weights and biases are initialised from $U\left(-1/\sqrt{\text{fan-in}}, 1/\sqrt{\text{fan-in}}\right)$, the default of PyTorch linear layers, and a fresh initialisation is drawn for every run. Training uses Adam [28] with learning rate 10^{-3}, $(\beta_1, \beta_2) = (0.9, 0.999)$, mini-batches of 256 and 100 epochs, with no hyperparameter tuning, so that differences between conditions are attributable to the loss and the contamination. The network and its gradients are implemented directly in NumPy so that every loss in Table 1 is applied through its score function ψ in the backward pass.

Losses whose scale or threshold depends on the residuals (adaptive Huber, Cauchy, Tukey and trimmed) are non-convex or scale-dependent and need a robust start, as in classical MM-estimation [11, 13]. They are therefore trained with the LAD loss for the first 10 epochs, after which the robust scale $\hat{\sigma} = 1.4826\,\text{MAD}$ of the training residuals is recomputed at the start of every epoch. This warm start is part of the estimator definition and is used under every condition, including clean data.

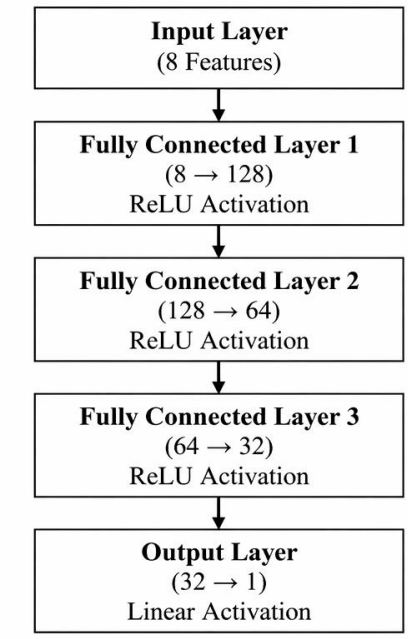


Figure 2. *Architecture of the 128-64-32 ReLU multilayer perceptron used for all DNN estimators.*

4.3 Contamination mechanisms

At each fraction f, a random subset of $\lfloor f \cdot 16{,}512 \rfloor$ training labels is selected without replacement and corrupted by one of three mechanisms:

- **Symmetric Cauchy (main experiment):** $Y_i \leftarrow Y_i + 50\,C_i$ with C_i standard Cauchy, for $f \in \{0{,}0.05, \ldots, 0.50\}$ (11 levels). This reproduces the protocol of the earlier version of the study.
- **One-sided gross errors:** $Y_i \leftarrow Y_i + 50\,|C_i|$, for $f \in \{0.1{,}0.2{,}0.3{,}0.4\}$.
- **Clustered contamination:** $Y_i \leftarrow 25$ (five times the largest genuine value), for $f \in \{0.1{,}0.2{,}0.3{,}0.4\}$.

For a given seed and fraction, the same corrupted label vector is used for all losses, so that comparisons between losses are paired.

4.4 Monte Carlo design

To measure estimation error against a known truth, we simulate the Friedman #1 function [33] with $d = 10$ uniform covariates, of which five are irrelevant:

$$f^*(x) = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5. \qquad (8)$$

Responses are $Y = f^*(X) + \varepsilon$ with $n = 2{,}000$ and seven error laws: Gaussian, Student-t_3, Student-t_2, symmetric α-stable with $\alpha = 1.5$ (generated by the method of Chambers, Mallows and Stuck [34]), symmetric Pareto with tail index 1.5, the contaminated normal $0.9\,N(0{,}1) + 0.1\,N(0, 10^2)$, and Cauchy. Each configuration is replicated 10 times. The same architecture (with 10 inputs) is trained with Adam for 300 epochs using mini-batches of 128, and with a 30-epoch LAD warm start for the scale-dependent losses.

4.5 Additional experiments

- **Sensitivity curve.** On a random subset of 2,000 training observations, $m \in \{1{,}20\}$ labels are shifted by $M \in \{10, 10^2, \ldots, 10^6\}$, and the networks are trained for 100 epochs with mini-batches of 64 (3 seeds; MSE, Huber, LAD and Tukey).
- **Training-choice ablation.** At 10% and 30% symmetric Cauchy contamination, MSE and Huber networks are trained plain, with L2 weight decay ($\lambda = 10^{-3}$), with dropout ($p = 0.2$), with gradient-norm clipping at 1, and with early stopping. For early stopping, 10% of the (contaminated) training set is held out and scored by the median absolute error, which is itself robust, with patience 10 (3 seeds).
- **Benchmarks.** Ordinary least squares; linear Huber regression ($\epsilon = 1.345$); linear LAD (median) regression [35]; a linear Tukey-bisquare M-estimator computed by IRLS from the LAD start [36]; random forests (200 trees, minimum leaf size 2) [37]; and histogram gradient boosting [38] with squared and absolute-error losses [32], all under the symmetric Cauchy protocol (3 seeds).

Table 2. *Summary of the experimental design.*

Experiment	Data	Factors	Replications	Runs
E1: Breakdown curves	California Housing	8 losses × 11 levels (symmetric Cauchy)	5 seeds	440
E2: Asymmetric contamination	California Housing	8 losses × 4 levels × 2 mechanisms	3 seeds	192

Experiment	Data	Factors	Replications	Runs
E3: Sensitivity curve	California subset (n = 2,000)	4 losses × 2 values of m × 7 values of M	3 seeds	168
E4: Monte Carlo	Friedman #1 (n = 2,000)	8 losses × 7 noise laws	10 replications	560
E5: Training choices	California Housing	2 losses × 5 variants × 2 levels	3 seeds	60
E6: Benchmarks	California Housing	7 estimators × 11 levels	3 seeds	231

5. Results and Discussion

5.1 Breakdown curves under symmetric Cauchy contamination

Figure 3 and Table 3 report clean test MAE for the eight losses across the eleven contamination levels. On clean data the losses are close: LAD attains 0.334, the Tukey, Cauchy and adaptive Huber losses 0.337, the Huber loss 0.346 and MSE 0.356 (×$100k). The trimmed loss is the exception, at 0.387, because discarding half of each mini-batch costs efficiency when there are no outliers to discard, the classical price of LTS-type estimators [5].

The MSE network breaks down at the first contamination level in every one of the five seeds. With 5% corrupted labels its mean test MAE rises to 17.8 (SD 8.5), a 50-fold increase over its clean value, and it fluctuates between about 45 and 125 at higher levels with very large seed-to-seed variability (coefficient of variation between 0.27 and 1.03 across levels). The non-monotone shape of the MSE curve reported in the earlier single-run analysis is therefore a property of the randomness of Cauchy draws and of the optimiser, not of the contamination level, and replicated runs are required to describe it.

All seven bounded-ψ objectives remain far below the breakdown threshold up to 45% contamination. At 50%, the LAD, Tukey and trimmed losses increase by only 14%, 21% and −4% relative to their clean values, compared with 31% for Huber, 36% for Cauchy and 40% for log-cosh. The adaptive Huber loss is the only robust loss that breaks down, at 45% (MAE 0.546) and severely at 50% (1.092). The reason is the scale estimate: when almost half of the residuals are gross errors, the MAD of the residuals, and hence the data-driven threshold $\delta = 1.345\hat{\sigma}$, inflates, so the loss gradually loses its robustness. A fixed threshold, or a scale fitted on a trusted subset, avoids this failure.

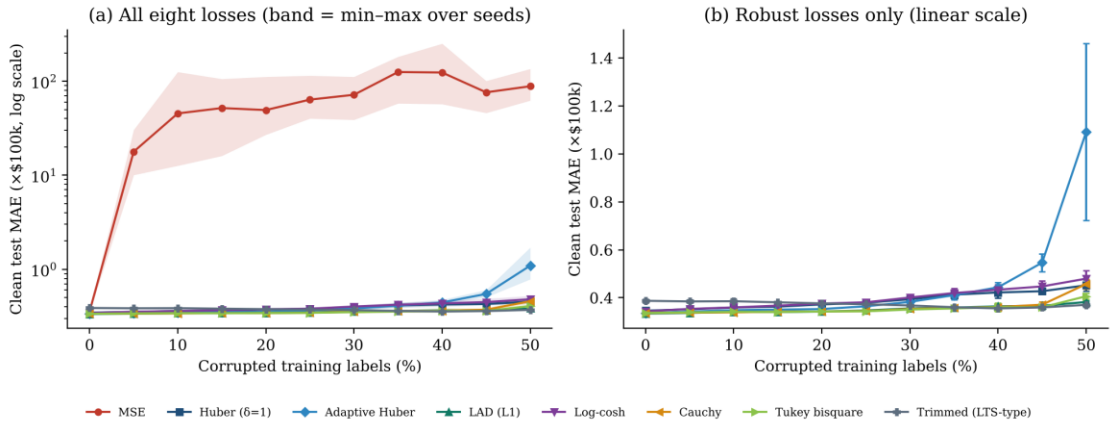


Figure 3. *Clean test MAE against the fraction of Cauchy-corrupted training labels (5 seeds). (a) All losses on a log scale; shaded bands show the minimum–maximum range over seeds. (b) Robust losses only, on a linear scale, mean ± SD.*

Table 3. *Clean test MAE (×$100k, mean ± SD over 5 seeds) on California Housing under symmetric Cauchy label contamination (γ = 50), and empirical breakdown point (Emp. BP: smallest tested level at which mean MAE exceeds 1.5 × the loss's own clean MAE).*

Loss	0%	5%	10%	20%	30%	40%	50%	Emp. BP
MSE	0.356 ± 0.006	17.8 ± 8.5	45.4 ± 47.0	49.3 ± 36.0	71.7 ± 30.7	123.7 ± 86.2	88.3 ± 30.3	5%
Huber (δ = 1)	0.346 ± 0.004	0.352 ± 0.006	0.359 ± 0.010	0.372 ± 0.007	0.394 ± 0.014	0.422 ± 0.023	0.452 ± 0.024	> 50%
Adaptive Huber	0.337 ± 0.005	0.342 ± 0.005	0.348 ± 0.006	0.353 ± 0.005	0.383 ± 0.010	0.443 ± 0.021	1.092 ± 0.368	45%
LAD (L1)	0.334 ± 0.005	0.337 ± 0.002	0.341 ± 0.002	0.342 ± 0.002	0.355 ± 0.009	0.364 ± 0.008	0.382 ± 0.006	> 50%
Log-cosh	0.343 ± 0.004	0.353 ± 0.004	0.360 ± 0.008	0.376 ± 0.005	0.401 ± 0.012	0.433 ± 0.022	0.480 ± 0.035	> 50%
Cauchy	0.337 ± 0.006	0.338 ± 0.006	0.339 ± 0.004	0.343 ± 0.004	0.353 ± 0.009	0.361 ± 0.013	0.457 ± 0.020	> 50%
Tukey bisquare	0.337 ± 0.004	0.339 ± 0.005	0.341 ± 0.001	0.343 ± 0.003	0.351 ± 0.008	0.360 ± 0.019	0.406 ± 0.012	> 50%
Trimmed (LTS-type)	0.387 ± 0.008	0.385 ± 0.007	0.385 ± 0.011	0.376 ± 0.006	0.368 ± 0.003	0.355 ± 0.002	0.370 ± 0.010	> 50%

5.2 Relative degradation

Figure 4 expresses each result relative to the loss's own clean baseline. MSE degrades by between 4,903% (5%) and 35,090% (35%). Among robust losses, Tukey, Cauchy and LAD stay within about 10% of their clean values up to 45% contamination, while the Huber and log-cosh losses already lose 14–17% at 30% contamination. The trimmed loss improves slightly as contamination grows, because the fraction of genuinely useful observations it discards decreases.

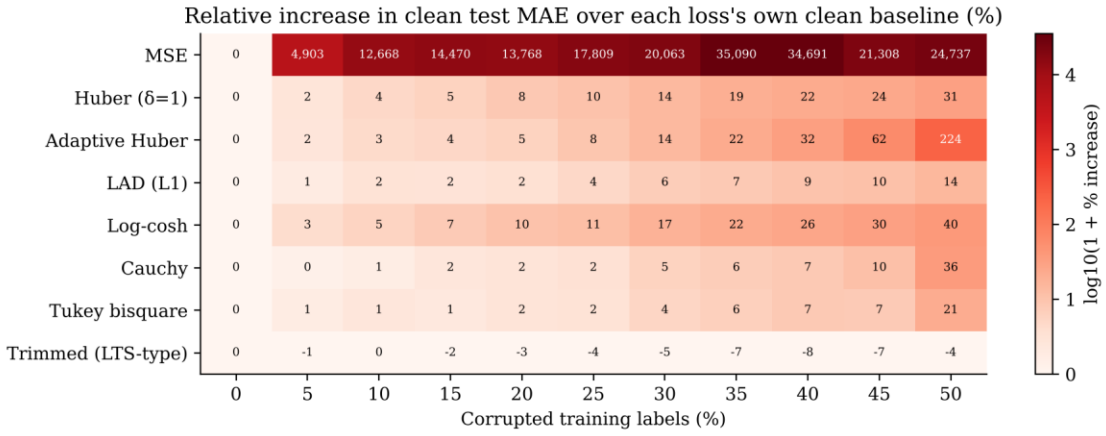


Figure 4. *Percentage increase of clean test MAE over each loss's own clean baseline (mean of 5 seeds).*

5.3 Statistical comparison of the losses

Table 4 and Figure 5 summarise the paired comparison over the 50 contaminated blocks (10 levels × 5 seeds). The Friedman test rejects equality of the eight losses decisively. The Tukey bisquare (mean rank 2.14), LAD (2.36) and Cauchy (2.54) losses form the leading group; their mutual differences are smaller than the Nemenyi critical difference of 1.48, so they are statistically indistinguishable. All three are significantly better than the fixed-threshold Huber loss (rank 5.38) by both the Nemenyi criterion and Holm-adjusted Wilcoxon tests. The trimmed loss is also significantly better than Huber, but by a small margin. The adaptive Huber loss is not significantly different from the Huber loss, and log-cosh is significantly worse. MSE ranks last in every block.

Table 4. *Statistical comparison over N = 50 blocks (10 contamination levels × 5 seeds). Friedman $\chi^2(7) = 249.0$, $p = 4.5e\text{-}50$; Nemenyi critical difference CD = 1.48 ($\alpha = 0.05$).*

Loss	Mean Friedman rank	Median ΔMAE vs Huber	Holm-adjusted p (Wilcoxon)	Verdict vs Huber
Tukey bisquare	2.14	-0.0340	$< 10^{-10}$	better
LAD (L1)	2.36	-0.0360	$< 10^{-10}$	better
Cauchy	2.54	-0.0292	$< 10^{-10}$	better
Trimmed (LTS-type)	4.40	-0.0093	0.0084	better

Loss	Mean Friedman rank	Median ΔMAE vs Huber	Holm-adjusted p (Wilcoxon)	Verdict vs Huber
Adaptive Huber	5.10	-0.0070	0.88	n.s.
Huber (δ = 1)	5.38	reference	—	—
Log-cosh	6.08	+0.0059	2.9e-07	worse
MSE	8.00	+59.07	$< 10^{-10}$	worse

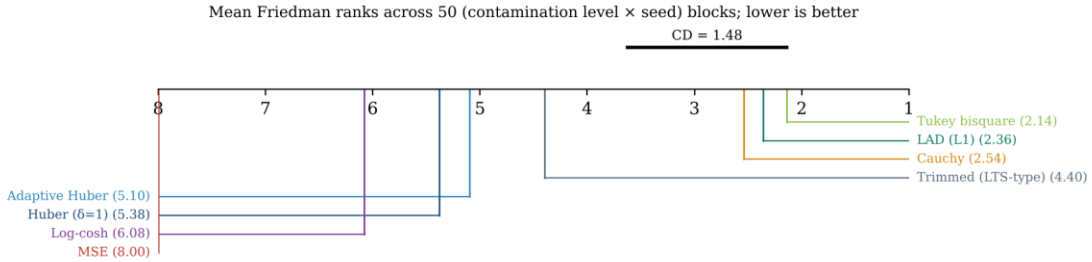


Figure 5. *Critical-difference diagram of mean Friedman ranks (lower is better). Losses closer than CD = 1.48 are not significantly different at the 5% level.*

5.4 Sensitivity curves: empirical verification of Propositions 1–3

Figure 6 and Table 5 show how far the trained network moves when m labels are shifted by M. For MSE the shift grows almost linearly in M, as Proposition 1 predicts for the stationary points. A single corrupted label of magnitude 10^6 in 2,000 moves the predictions by 471 on average, and 20 such labels move them by 11,561. The bounded-ψ networks behave exactly as Propositions 2 and 3 predict. Their sensitivity curves are perfectly flat from M = 10 onwards: once the corrupted residuals are in the saturation region, the trained network is identical, to every reported digit, whatever the magnitude of the outlier. The remaining shift (0.05–0.09) is smaller than the change produced simply by retraining on clean data with a different seed (0.19).

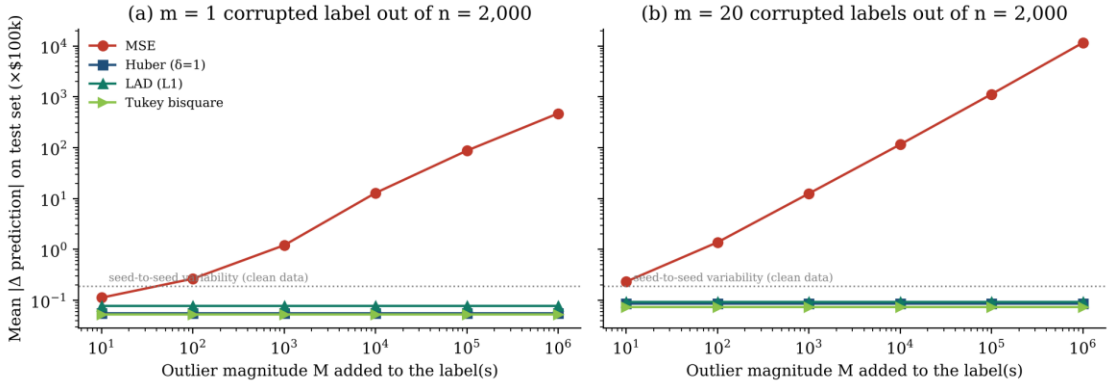


Figure 6. *Sensitivity curves: mean absolute change in test predictions when m labels are shifted by M (3 seeds). The dotted line marks the average change between two clean-data runs that differ only in the random seed.*

Table 5. *Sensitivity curve: mean absolute change in test-set predictions when m of n = 2,000 training labels are shifted by +M (mean of 3 seeds, ×$100k). For reference, retraining on clean data with a different seed changes predictions by 0.190 on average.*

m	Loss	M = 10	10^2	10^3	10^4	10^5	10^6
1	MSE	0.113	0.266	1.208	12.881	87.432	471
1	Huber (δ = 1)	0.056	0.056	0.056	0.056	0.056	0.056
1	LAD (L1)	0.078	0.078	0.078	0.078	0.078	0.078
1	Tukey bisquare	0.053	0.053	0.053	0.053	0.053	0.053
20	MSE	0.233	1.384	12.568	116	1,128	11,561
20	Huber (δ = 1)	0.086	0.086	0.086	0.086	0.086	0.086
20	LAD (L1)	0.093	0.093	0.093	0.093	0.093	0.093
20	Tukey bisquare	0.074	0.074	0.074	0.074	0.074	0.074

5.5 Asymmetric contamination separates monotone from redescending losses

The symmetric Cauchy protocol favours monotone bounded losses, because the constant pushes of positive and negative outliers largely cancel (Proposition 3). Figure 7 and Tables 6–7 show what happens when they do not. Under one-sided gross errors, the Huber and log-cosh losses break down at 20% contamination and LAD at 40%; under clustered contamination at $y = 25$, Huber breaks down at 30%, log-cosh at 20% and LAD at 40%. The adaptive Huber loss breaks down at 30% under both mechanisms. In contrast, the Tukey bisquare loss, which sets ψ to zero for gross residuals, stays between 0.34 and 0.38 at all levels under both mechanisms (at most +14% at 40% one-sided contamination), and the trimmed loss is essentially unaffected. The Cauchy loss, whose score decreases only slowly, is intermediate: robust throughout, but with a 36–40% increase at 40%. This is the regression analogue of the classical distinction between monotone M-estimators and redescending or high-breakdown estimators [5, 13], and it is the most important practical finding of the study. Robustness to symmetric heavy-tailed noise does not imply robustness to systematic gross errors.

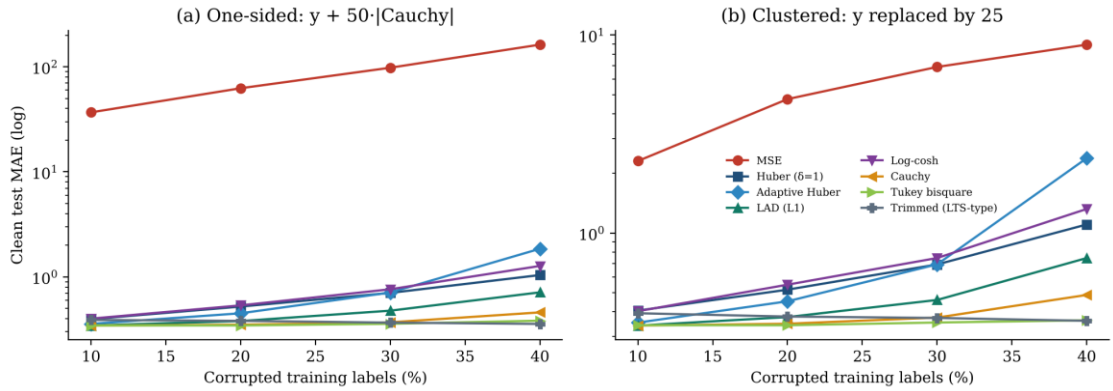


Figure 7. *Clean test MAE under (a) one-sided gross errors and (b) clustered contamination (3 seeds; log scale).*

Table 6. *Clean test MAE (mean ± SD, 3 seeds) under asymmetric contamination. OS = one-sided gross errors (y + 50·|Cauchy|); CL = clustered replacement (y := 25).*

Loss	OS 10%	OS 20%	OS 30%	OS 40%	CL 10%	CL 20%	CL 30%	CL 40%
MSE	36.8 ± 13.9	62.5 ± 25.6	98.0 ± 32.4	162.6 ± 80.7	2.310 ± 0.102	4.743 ± 0.196	6.895 ± 0.118	8.935 ± 0.285
Huber (δ = 1)	0.395 ± 0.030	0.521 ± 0.013	0.703 ± 0.063	1.040 ± 0.084	0.406 ± 0.027	0.518 ± 0.013	0.694 ± 0.048	1.105 ± 0.108
Adaptive Huber	0.357 ± 0.007	0.450 ± 0.010	0.710 ± 0.063	1.852 ± 0.017	0.353 ± 0.011	0.452 ± 0.006	0.695 ± 0.056	2.388 ± 0.338
LAD (L1)	0.345 ± 0.001	0.380 ± 0.014	0.476 ± 0.043	0.712 ± 0.098	0.341 ± 0.003	0.376 ± 0.017	0.459 ± 0.028	0.748 ± 0.110
Log-cosh	0.400 ± 0.025	0.535 ± 0.031	0.762 ± 0.095	1.270 ± 0.075	0.403 ± 0.028	0.549 ± 0.032	0.748 ± 0.067	1.323 ± 0.098
Cauchy	0.343 ± 0.003	0.351 ± 0.004	0.369 ± 0.015	0.459 ± 0.027	0.341 ± 0.003	0.348 ± 0.005	0.374 ± 0.021	0.487 ± 0.030
Tukey bisquare	0.344 ± 0.002	0.344 ± 0.005	0.357 ± 0.013	0.383 ± 0.027	0.342 ± 0.003	0.342 ± 0.001	0.353 ± 0.012	0.363 ± 0.009
Trimmed (LTS-type)	0.389 ± 0.002	0.381 ± 0.007	0.367 ± 0.003	0.356 ± 0.007	0.394 ± 0.007	0.379 ± 0.011	0.373 ± 0.007	0.361 ± 0.002

Table 7. *Empirical breakdown points (1.5 × clean-MAE criterion) of the DNN estimator for the three contamination mechanisms.*

Loss	Symmetric Cauchy	One-sided	Clustered	ψ type
MSE	5%	≤ 10%	≤ 10%	unbounded
Huber (δ = 1)	> 50%	20%	30%	monotone, bounded
Adaptive Huber	45%	30%	30%	monotone, bounded (data-driven δ)
LAD (L1)	> 50%	40%	40%	monotone, bounded
Log-cosh	> 50%	20%	20%	monotone, bounded
Cauchy	> 50%	> 40%	> 40%	redescending (slowly)
Tukey bisquare	> 50%	> 40%	> 40%	redescending (to zero)
Trimmed (LTS-type)	> 50%	> 40%	> 40%	hard rejection (top 50%)

5.6 Monte Carlo study with heavy-tailed noise

Table 8 and Figure 8 report the error in estimating the true function f^*. Under Gaussian noise all losses except the trimmed one perform similarly (0.91–1.03), which shows that the robust losses cost little efficiency in this setting. In this design the LAD network is even slightly more accurate than MSE (0.908 vs 0.952), plausibly because its bounded gradient acts as an implicit regulariser over 300 untuned epochs. As the tails become heavier, the MSE error grows steadily: from 0.95 (Gaussian) to 1.49 (t_3), 2.51 (t_2), 2.77 (contaminated normal), 3.72 (α-stable 1.5), 7.36 (Pareto 1.5) and 37.5 under Cauchy noise, where its mean is dominated by occasional catastrophic replications (median 17.8). The robust losses degrade far more slowly. LAD is the most accurate loss for the moderately heavy tails (t_3, t_2, α-stable), LAD and Tukey bisquare are tied under Pareto noise (1.120 vs 1.121), and Tukey bisquare is the most accurate under the contaminated normal and Cauchy laws (1.50 under Cauchy noise against 1.98 for Huber). The 50%-trimmed loss is consistently inefficient (about 1.7–1.9) because it discards half of the clean information. Over the 60 non-Gaussian blocks, the Friedman test again rejects equality of the losses ($\chi^2(7) = 339.6$, $p < 10^{-60}$).

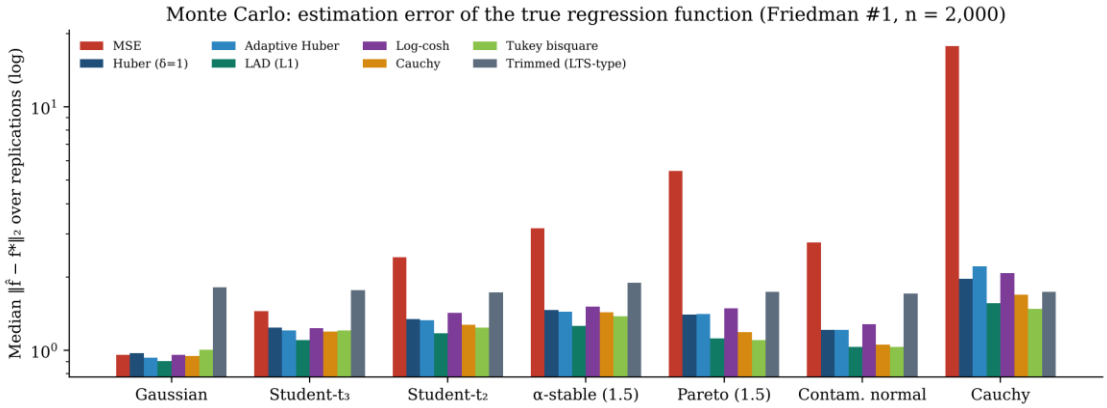


Figure 8. *Monte Carlo study: median estimation error of the true regression function across 10 replications for seven noise laws (log scale).*

Table 8. *Monte Carlo study (Friedman #1 function, d = 10, n = 2,000, 10 replications): mean L2 estimation error ‖f̂ − f*‖ on 5,000 noise-free test points (standard error in parentheses). Best value per row in bold.*

Noise	MSE	Huber (δ = 1)	Adaptive Huber	LAD (L1)	Log-cosh	Cauchy	Tukey bisquare	Trimmed (LTS-type)
Gaussian N(0,1)	0.952 (0.014)	0.972 (0.012)	0.944 (0.011)	**0.908** (0.013)	0.974 (0.018)	0.948 (0.011)	1.029 (0.024)	1.808 (0.067)
Student-t_3	1.490 (0.051)	1.232 (0.013)	1.196 (0.019)	**1.098** (0.012)	1.229 (0.012)	1.182 (0.015)	1.206 (0.020)	1.725 (0.050)
Student-t_2	2.510 (0.138)	1.367 (0.031)	1.350 (0.026)	**1.186** (0.023)	1.426 (0.029)	1.271 (0.023)	1.247 (0.020)	1.729 (0.048)
Symmetric α-stable, α = 1.5	3.722 (0.374)	1.481 (0.029)	1.475 (0.034)	**1.276** (0.021)	1.537 (0.040)	1.403 (0.022)	1.383 (0.016)	1.892 (0.055)
Symmetric Pareto, α = 1.5	7.359 (1.398)	1.400 (0.041)	1.425 (0.021)	**1.120** (0.010)	1.472 (0.030)	1.201 (0.019)	1.121 (0.022)	1.835 (0.084)
0.9 N(0,1) + 0.1 N(0,10^2)	2.773 (0.074)	1.240 (0.046)	1.201 (0.030)	1.061 (0.027)	1.291 (0.032)	1.067 (0.020)	**1.046** (0.014)	1.700 (0.072)
Cauchy (α = 1)	37.482 (14.589)	1.981 (0.060)	2.235 (0.081)	1.569 (0.031)	2.106 (0.071)	1.706 (0.031)	**1.500** (0.027)	1.736 (0.044)

5.7 Do common training choices protect the MSE network?

Table 9 and Figure 9 test whether implicit robustness mechanisms rescue MSE training. Weight decay does not materially help. Dropout and gradient-norm clipping reduce the MSE error by factors of about 2 to 4 but leave it 19

to 93 times above its clean baseline. Early stopping with a robustly scored validation split is the most effective: it reduces the MSE error from 29.6 to 1.42 at 10% contamination and from 64.9 to 1.82 at 30%. This agrees with the theory that early-stopped gradient descent fits clean structure before memorising corrupted labels [26, 27]. Even so, the early-stopped MSE network remains above the breakdown threshold (four to five times its clean MAE) and is four times worse than a plain Huber network. For Huber training the same choices change the MAE by at most 0.03. Implicit regularisation is therefore not a substitute for a robust loss.

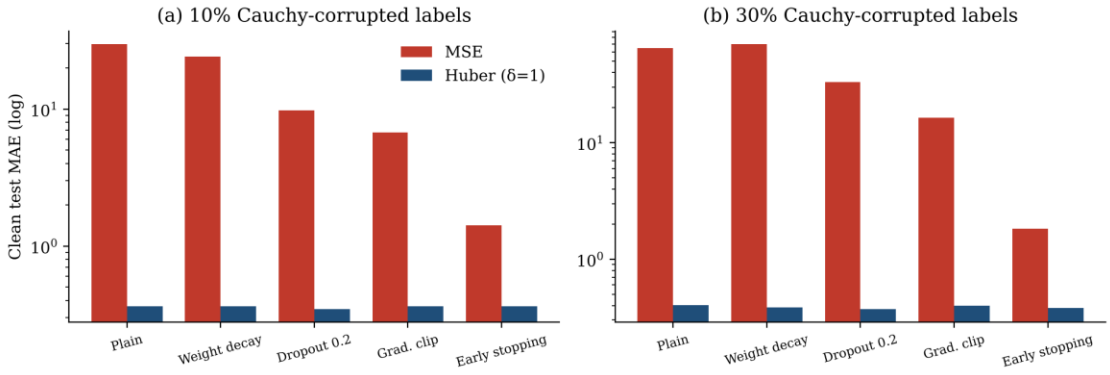


Figure 9. *Effect of training choices on clean test MAE at (a) 10% and (b) 30% Cauchy contamination (3 seeds; log scale).*

Table 9. *Effect of training choices on clean test MAE (mean ± SD, 3 seeds) under 10% and 30% symmetric Cauchy contamination.*

Training choice	MSE, 10%	Huber, 10%	MSE, 30%	Huber, 30%
Plain (no regularisation)	29.6 ± 17.1	0.364 ± 0.008	64.9 ± 40.6	0.401 ± 0.010
Weight decay ($\lambda = 10^{-3}$)	24.1 ± 8.0	0.364 ± 0.003	69.8 ± 53.8	0.387 ± 0.022
Dropout (p = 0.2)	9.798 ± 3.478	0.346 ± 0.002	33.2 ± 20.0	0.372 ± 0.002
Gradient-norm clipping ($\|g\| \leq 1$)	6.731 ± 1.497	0.362 ± 0.004	16.3 ± 0.6	0.398 ± 0.007
Early stopping (robust validation, patience 10)	1.418 ± 0.390	0.364 ± 0.008	1.818 ± 0.249	0.382 ± 0.009

5.8 Comparison with classical and ensemble estimators

Table 10 and Figure 10 place the DNN results among standard alternatives. Every squared-error method breaks down at 5% contamination: OLS, random forests and gradient boosting with squared loss reach test MAEs in the tens to hundreds. The linear robust M-estimators (Huber, LAD, Tukey) never break down, but their baseline error is about 0.51 because a linear model cannot capture the nonlinear price surface. Gradient boosting with the absolute-error loss is the strongest competitor: it has the lowest clean error (0.316) and degrades only to 0.369 at 50% contamination, slightly better than the best DNN losses (LAD 0.382 and Tukey 0.406 at 50%). Robustness is therefore a property of the loss rather than of the model class. On this tabular dataset, a robust DNN is competitive with, but not superior to, robust gradient boosting.

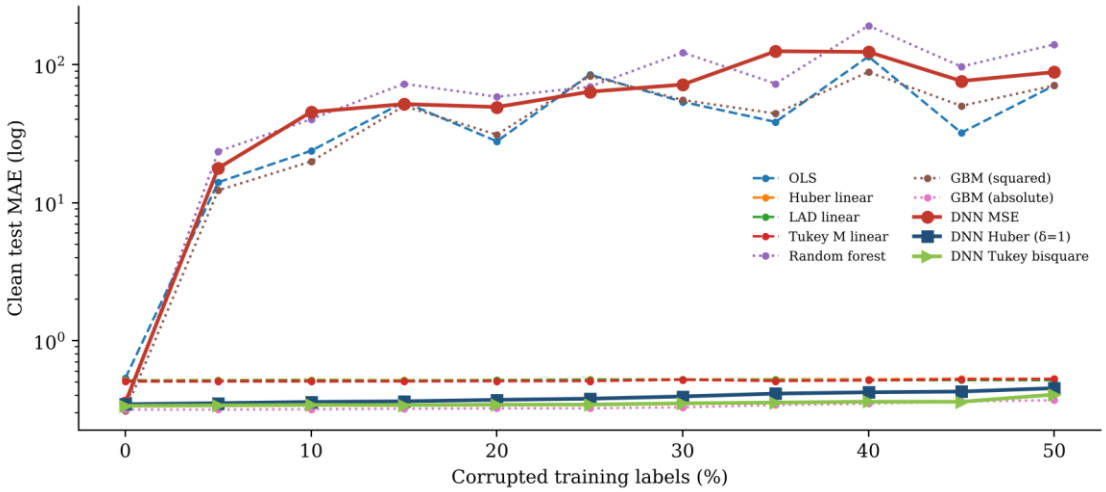


Figure 10. *Clean test MAE of benchmark estimators and selected DNN losses under symmetric Cauchy contamination (log scale).*

Table 10. *Benchmark comparison under symmetric Cauchy contamination: clean test MAE (mean ± SD; 3 seeds for benchmarks, 5 for DNNs).*

Estimator	0%	10%	30%	50%	Emp. BP
OLS	0.534 ± 0.000	23.7 ± 15.0	53.7 ± 46.1	70.7 ± 36.1	5%
Huber linear	0.516 ± 0.000	0.520 ± 0.000	0.524 ± 0.001	0.529 ± 0.014	> 50%
LAD linear	0.512 ± 0.000	0.515 ± 0.000	0.521 ± 0.000	0.516 ± 0.011	> 50%
Tukey M linear	0.505 ± 0.000	0.506 ± 0.000	0.521 ± 0.010	0.527 ± 0.015	> 50%
Random forest	0.325 ± 0.000	40.2 ± 23.9	122.5 ± 87.1	140.3 ± 50.3	5%
GBM (squared)	0.310 ± 0.000	19.9 ± 0.9	55.5 ± 32.9	70.6 ± 30.3	5%
GBM (absolute)	0.316 ± 0.000	0.318 ± 0.002	0.328 ± 0.004	0.369 ± 0.002	> 50%
DNN, MSE	0.356 ± 0.006	45.4 ± 47.0	71.7 ± 30.7	88.3 ± 30.3	5%
DNN, Huber (δ = 1)	0.346 ± 0.004	0.359 ± 0.010	0.394 ± 0.014	0.452 ± 0.024	> 50%
DNN, LAD (L1)	0.334 ± 0.005	0.341 ± 0.002	0.355 ± 0.009	0.382 ± 0.006	> 50%
DNN, Tukey bisquare	0.337 ± 0.004	0.341 ± 0.001	0.351 ± 0.008	0.406 ± 0.012	> 50%

5.9 Comparison with related work

Table 11 positions the study relative to the closest prior work. Werner [21] provided the breakdown framework for feed-forward networks and simulation evidence that robust losses help. The present study adds heavy-tailed and asymmetric contamination, a statistically replicated real-data benchmark, redescending and trimmed objectives, sensitivity curves and benchmarks. Relative to the theory of Shen et al. [8], Fan et al. [9] and Feng and Wu [19], it supplies the breakdown-type evidence that these rate results do not address.

Table 11. *Comparison with related work.*

Study	Model	Contamination / noise	Real data	Losses compared	Replication and tests
Werner [21]	Feed-forward NN	Simulated contamination	No	MSE and robust losses	Simulation
Tyralis et al. [24]	Deep Huber quantile NN	Natural noise	Yes	Huber quantile	—
Song et al. [6]; Chen et al. [25]	Various DNNs	Label flipping (classification)	Yes	Many	Survey
Shen et al. [8]; Fan et al. [9]; Feng & Wu [19]	DNN / nonparametric	Heavy-tailed noise (theory)	No	Huber, LAD, others	Theory
Present study	128-64-32 ReLU MLP	Symmetric Cauchy, one-sided, clustered; 7 noise laws	Yes (California Housing)	8 losses + 7 benchmarks	5 seeds; Friedman, Nemenyi, Wilcoxon–Holm

6. Practical Implications and Limitations

The results lead to four recommendations for DNN regression on data that may contain gross errors. First, the squared-error loss should not be used when even a few percent of the labels may be grossly wrong. Its breakdown is immediate, it occurred in every seed, and regularisation, dropout and clipping do not prevent it. Second, robustness to symmetric heavy-tailed noise is a weak requirement that any bounded score satisfies. The discriminating case is asymmetric or clustered contamination, where redescending (Tukey) or trimmed objectives are needed. Third, scale-dependent losses require a robust warm start and a robust scale. A short LAD phase followed by a MAD scale works well up to about 40% contamination, but a data-driven Huber threshold computed from contaminated residuals fails near 50%. Fourth, when no robust loss can be used, early stopping on a robustly scored validation set is the most effective mitigation among those studied.

The study has limitations. Contamination affects responses only; leverage contamination of the covariates, against which monotone M-estimators are known to fail [5], was not studied. A single real dataset and a single architecture

were used, and hyperparameters were deliberately fixed rather than tuned per loss, which may disadvantage some losses (for example, a tuned trimming fraction would reduce the efficiency loss of the trimmed objective). The number of seeds (3–5 per condition) was limited by computation, although the paired design and rank-based tests make the main conclusions statistically clear. Extending the analysis to leverage contamination, other architectures and naturally heavy-tailed applications such as financial returns is left for future work.

7. Conclusions

This paper revisited the breakdown point of DNN regression estimators under heavy-tailed noise, with theory, replicated experiments and benchmarks. Theoretically, MSE stationary points have breakdown point 1/n, gradient training with a bounded score yields networks that remain bounded under any fraction of label contamination, and bounded and redescending losses become exactly invariant to outlier magnitude once residuals saturate. Because the classical breakdown point cannot rank robust losses for DNNs, we compared them with empirical breakdown points, sensitivity curves and rank-based tests. Empirically, the MSE-trained network breaks down at 5% contamination in every run, with a 50-fold increase in test error. All bounded losses survive 45% symmetric Cauchy contamination, but only the redescending and trimmed objectives survive one-sided and clustered contamination up to 40%. The Tukey bisquare loss with a robust warm start achieved the best overall rank on real data and the lowest error under the heaviest-tailed simulated noise, and LAD was best for moderately heavy tails. Gradient boosting with an absolute-error loss was equally robust, which confirms that the loss, not the model class, determines robustness. Future work will extend the analysis to leverage contamination, deeper architectures and naturally heavy-tailed financial data, and will study the theoretical breakdown properties of redescending DNN estimators.

Data and code availability. All experiments use publicly available data and fixed random seeds. The complete code (network implementation, experiment runner, analysis and figure scripts) and the raw result files for every run are provided as supplementary material.

References

[1] F. R. Hampel, “A general qualitative definition of robustness,” The Annals of Mathematical Statistics, vol. 42, no. 6, pp. 1887–1896, 1971, doi: 10.1214/aoms/1177693054.

[2] P. J. Huber, “Robust estimation of a location parameter,” The Annals of Mathematical Statistics, vol. 35, no. 1, pp. 73–101, 1964, doi: 10.1214/aoms/1177703732.

[3] D. L. Donoho and P. J. Huber, “The notion of breakdown point,” in A Festschrift for Erich L. Lehmann, P. J. Bickel, K. Doksum, and J. L. Hodges, Eds. Belmont, CA, USA: Wadsworth, 1983, pp. 157–184.

[4] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, Robust Statistics: The Approach Based on Influence Functions. New York, NY, USA: Wiley, 1986.

[5] P. J. Rousseeuw and A. M. Leroy, Robust Regression and Outlier Detection. New York, NY, USA: Wiley, 1987.

[6] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, “Learning from noisy labels with deep neural networks: A survey,” IEEE Transactions on Neural Networks and Learning Systems, vol. 34, no. 11, pp. 8135–8153, Nov. 2023, doi: 10.1109/TNNLS.2022.3152527.

[7] J. Schmidt-Hieber, “Nonparametric regression using deep neural networks with ReLU activation function,” The Annals of Statistics, vol. 48, no. 4, pp. 1875–1897, Aug. 2020, doi: 10.1214/19-AOS1875.

[8] G. Shen, Y. Jiao, Y. Lin, and J. Huang, "Robust nonparametric regression with deep neural networks," arXiv preprint arXiv:2107.10343, Jul. 2021.

[9] J. Fan, Y. Gu, and W.-X. Zhou, "How do noise tails impact on deep ReLU networks?" The Annals of Statistics, vol. 52, no. 4, pp. 1845–1871, Aug. 2024, doi: 10.1214/24-AOS2428.

[10] P. J. Rousseeuw, "Least median of squares regression," Journal of the American Statistical Association, vol. 79, no. 388, pp. 871–880, 1984, doi: 10.1080/01621459.1984.10477105.

[11] V. J. Yohai, "High breakdown-point and high efficiency robust estimates for regression," The Annals of Statistics, vol. 15, no. 2, pp. 642–656, 1987, doi: 10.1214/aos/1176350366.

[12] A. E. Beaton and J. W. Tukey, "The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data," Technometrics, vol. 16, no. 2, pp. 147–185, 1974, doi: 10.1080/00401706.1974.10489171.

[13] R. A. Maronna, R. D. Martin, V. J. Yohai, and M. Salibián-Barrera, Robust Statistics: Theory and Methods (with R), 2nd ed. Hoboken, NJ, USA: Wiley, 2019.

[14] Y. Jiao, G. Shen, Y. Lin, and J. Huang, "Deep nonparametric regression on approximate manifolds: Nonasymptotic error bounds with polynomial prefactors," The Annals of Statistics, vol. 51, no. 2, pp. 691–716, Apr. 2023, doi: 10.1214/23-AOS2266.

[15] J. Chen, "Robust nonparametric regression based on deep ReLU neural networks," Statistics & Probability Letters, vol. 208, p. 110046, 2024, doi: 10.1016/j.spl.2024.110046.

[16] H. Xu, Y. An, and J. Lu, "Robust learning of Huber loss under weak conditional moment," Neurocomputing, vol. 507, pp. 191–201, 2022, doi: 10.1016/j.neucom.2022.08.012.

[17] T. Liu, Z. Xu, and Q. Ling, "Functional linear regression with Huber loss," Journal of Complexity, vol. 74, p. 101696, 2023, doi: 10.1016/j.jco.2022.101696.

[18] S. Chen, F. Koehler, A. Moitra, and M. Yau, "Online and distribution-free robustness: Regression and contextual bandits with Huber contamination," in Proc. 62nd IEEE Annual Symposium on Foundations of Computer Science (FOCS), 2022, pp. 684–695, doi: 10.1109/FOCS52979.2021.00070.

[19] Y. Feng and Q. Wu, "Understanding robust machine learning for nonparametric regression with heavy-tailed noise," arXiv preprint arXiv:2510.09888, Oct. 2025.

[20] J. T. Barron, "A general and adaptive robust loss function," in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 4331–4339.

[21] T. Werner, "Global quantitative robustness of regression feed-forward neural networks," Neural Computing and Applications, vol. 36, pp. 19967–19988, 2024, doi: 10.1007/s00521-024-10289-w.

[22] A. Rubio and J. R. Dorronsoro, "Robust losses in deep regression," in Hybrid Artificial Intelligent Systems (HAIS 2023), Lecture Notes in Computer Science, vol. 14001. Cham, Switzerland: Springer, 2023, pp. 259–270, doi: 10.1007/978-3-031-40725-3_22.

[23] X. Wang, L. Zhou, and H. Lin, "Deep regression learning with optimal loss function," Journal of the American Statistical Association, vol. 120, no. 550, pp. 1305–1317, 2025, doi: 10.1080/01621459.2024.2412364.

[24] H. Tyralis, G. Papacharalampous, A. Langousis, and S. M. Papalexiou, "Deep Huber quantile regression networks," Neural Networks, vol. 186, p. 107364, 2025, doi: 10.1016/j.neunet.2025.107364.

[25] Y. Chen, Z. Wang, X. Liu, H. Li, and J. Zhang, "A survey on learning from data with label noise via deep neural networks," Systems Science & Control Engineering, vol. 13, no. 1, art. 2488120, 2025, doi: 10.1080/21642583.2025.2488120.

[26] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in Proc. 5th Int. Conf. Learning Representations (ICLR), 2017.

[27] M. Li, M. Soltanolkotabi, and S. Oymak, "Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks," in Proc. 23rd Int. Conf. Artificial Intelligence and Statistics (AISTATS), PMLR vol. 108, 2020, pp. 4313–4324.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proc. 3rd Int. Conf. Learning Representations (ICLR), 2015, arXiv:1412.6980.

[29] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," Journal of Machine Learning Research, vol. 7, pp. 1–30, 2006.

[30] S. Holm, "A simple sequentially rejective multiple test procedure," Scandinavian Journal of Statistics, vol. 6, no. 2, pp. 65–70, 1979.

[31] R. K. Pace and R. Barry, "Sparse spatial autoregressions," Statistics & Probability Letters, vol. 33, no. 3, pp. 291–297, 1997, doi: 10.1016/S0167-7152(96)00140-X.

[32] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[33] J. H. Friedman, "Multivariate adaptive regression splines," The Annals of Statistics, vol. 19, no. 1, pp. 1–67, 1991, doi: 10.1214/aos/1176347963.

[34] J. M. Chambers, C. L. Mallows, and B. W. Stuck, "A method for simulating stable random variables," Journal of the American Statistical Association, vol. 71, no. 354, pp. 340–344, 1976, doi: 10.1080/01621459.1976.10480344.

[35] R. Koenker and G. Bassett, "Regression quantiles," Econometrica, vol. 46, no. 1, pp. 33–50, 1978, doi: 10.2307/1913643.

[36] S. Seabold and J. Perktold, "statsmodels: Econometric and statistical modeling with Python," in Proc. 9th Python in Science Conference (SciPy), 2010, pp. 92–96.

[37] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[38] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," The Annals of Statistics, vol. 29, no. 5, pp. 1189–1232, 2001, doi: 10.1214/aos/1013203451.