Robust Regression Trees Based on the Maximum Correntropy Criterion

Goktug T. Cinar*, Ryan M. Burt*, Aslihan Demirkaya†, Jose C. Principe*
* Department of ECE, Computational NeuroEngineering Laboratory
University of Florida, Gainesville, FL, USA
goktugcinar@gmail.com, rburtufl@gmail.com, principe@cnel.ufl.edu
† Department of Mathematics
University of Hartford, West Hartford, CT, USA
demirkaya@hartford.edu

***Abstract*—Standard regression trees optimize squared loss and can be sensitive to outliers and heavy-tailed noise. We propose a Maximum Correntropy Regression Tree (MCRTree), in which the Maximum Correntropy Criterion (MCC) governs both leaf estimation and split selection. Each leaf prediction is obtained by a correntropy-weighted fixed-point update, yielding a robust estimator with redescending influence. We define the corresponding node impurity and split gain, and train the tree using a global bandwidth selected from a small MAD-based grid. On synthetic regression tasks, MCRTree matches standard MSE trees under Gaussian noise and improves absolute-error metrics under heavy-tailed contamination, in some cases slightly outperforming LAD trees. On the Beijing PM2.5 dataset, MCRTree improves median error relative to CART while remaining competitive with LAD. A bandwidth sensitivity study shows stable behavior across a broad range of kernel scales.**

I. Introduction

Decision trees have long been a cornerstone for machine learning in both classification and regression. For regression trees specifically, the CART algorithm minimizes mean square error (MSE) in each leaf and chooses splits that maximize the reduction in MSE [1] to produce the desired outcome. While this has been useful across many applications, it can be sensitive to outliers and heavy-tailed noise, leading to suboptimal trees under these conditions.

Robust statistics provide alternatives to quadratic loss, including Huber and least-absolute-deviation (LAD) regression [2], and several CART implementations support LAD-type impurity measures [3]. In the ensemble setting, quantile regression forests model conditional distributions and provide robustness via conditional quantiles [4]. In parallel, information-theoretic learning introduced correntropy and the Maximum Correntropy Criterion (MCC) as bounded kernel-based objectives for learning under non-Gaussian and heavy-tailed noise [5], [6]. These approaches provide robustness by downweighting large residuals, but have rarely been incorporated into the node-level objective of single regression trees.

In this work we propose a Maximum Correntropy Regression Tree (MCRTree), a single-tree regression method in which MCC serves as the common node-level objective for both robust leaf estimation and split selection. The resulting tree retains the interpretability of standard regression trees while being less sensitive to outliers and heavy-tailed noise.

Our contributions are as follows:

- We formulate a single-tree regression framework in which MCC governs both node prediction and split evaluation, and we derive the resulting node-wise leaf estimator as a fixed-point update with an M-estimation interpretation.
- We define an MCC-based node impurity and split gain that parallel the CART variance-reduction criterion while being driven by a bounded, localized similarity measure.
- We propose a practical training algorithm with bandwidth selection based on a MAD scale and a small grid over a single scalar factor.
- We empirically demonstrate on synthetic Gaussian and heavy-tailed regression problems and on the Beijing PM2.5 air-quality dataset [7] that MCRTree matches standard MSE trees under light-tailed noise and improves robustness under heavy-tailed noise. A bandwidth sensitivity analysis further illustrates how prediction quality varies with the kernel scale parameter.

Authors will make the code available upon publication.

II. Background: Correntropy and MCC

Let Y and $\hat{Y}$ be real-valued random variables. The correntropy between Y and $\hat{Y}$ with Gaussian kernel of bandwidth $\sigma > 0$ is defined as [5], [6]

$$V_\sigma(Y, \hat{Y}) = \mathbb{E}\left[\kappa_\sigma(Y - \hat{Y})\right], \tag{1}$$

where

$$\kappa_\sigma(e) = \exp\left(-\frac{e^2}{2\sigma^2}\right) \tag{2}$$

is the Gaussian kernel. Given a sample $\{(y_i, \hat{y}_i)\}_{i=1}^n$, an empirical estimate of correntropy is

$$\widehat{V}_\sigma = \frac{1}{n}\sum_{i=1}^{n} \exp\left(-\frac{(y_i - \hat{y}_i)^2}{2\sigma^2}\right). \tag{3}$$

The Maximum Correntropy Criterion (MCC) seeks an estimator that maximizes this empirical correntropy. For regression with model $f(x;\theta)$ and data $\{(x_i, y_i)\}$, the MCC objective is

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n} \kappa_\sigma(y_i - f(x_i;\theta)). \tag{4}$$

Maximizing $J(\theta)$ corresponds to minimizing an associated robust & bounded loss that downweights large residuals [5], [6]. When σ is small, the influence of outliers is strongly suppressed; as $\sigma \to \infty$, the MCC approximates quadratic loss. MCC behaves locally like a quadratic criterion around

small residuals. Consequently, when the noise is light-tailed and residuals remain concentrated near zero, MCC induces behavior close to standard least-squares fitting, whereas its bounded kernel suppresses the influence of large residuals in contaminated regimes.

III. Related Work

Regression trees are classically learned by variance reduction under squared loss [1], which makes them sensitive to outliers in the response. Earlier work on robust single-tree regression replaced least-squares fitting with robust M-estimation objectives such as Huber and Tukey-type criteria [8], and practical LAD-based implementations are now available, e.g., `rpart.LAD` [3]. Recent work has also explored outlier-aware tree modifications based on Huber-style fitting and residual-based outlier handling [9], [10].

Robustness has also been pursued in tree ensembles, for example through quantile regression forests [4] and robust Bayesian additive regression trees [11]. In parallel, information-theoretic learning introduced correntropy and MCC as bounded kernel-based objectives for learning under non-Gaussian noise [5], [6], and recent work continues to extend correntropy-based learning in robust regression and perception problems [12], [13]. In contrast to prior robust-tree work based mainly on L_1, Huber, or Tukey-type criteria, our contribution is to use MCC as a unified node-level objective for both leaf estimation and split selection in a single tree.

IV. Regression Trees and Node Impurity

Consider a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ with $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$. A regression tree partitions the input space into regions (leaves) via axis-aligned splits, and assigns a constant prediction to each leaf. In classical CART regression [1], the leaf value for a node with data indices $\mathcal{I}$ is the sample mean

$$\bar{y} = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} y_i, \tag{5}$$

and the node impurity is the mean squared error

$$I_{\mathrm{MSE}}(\mathcal{I}) = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} (y_i - \bar{y})^2. \tag{6}$$

For a candidate split s that partitions $\mathcal{I}$ into left and right subsets $\mathcal{I}_L$ and $\mathcal{I}_R$, the split quality is defined as the reduction in impurity:

$$\Delta I_{\mathrm{MSE}}(s) = I_{\mathrm{MSE}}(\mathcal{I}) - \left(\frac{|\mathcal{I}_L|}{|\mathcal{I}|} I_{\mathrm{MSE}}(\mathcal{I}_L) + \frac{|\mathcal{I}_R|}{|\mathcal{I}|} I_{\mathrm{MSE}}(\mathcal{I}_R) \right). \tag{7}$$

The algorithm chooses the split s with the largest ΔI_{MSE}.

V. Leaf Estimator under MCC

We now replace the MSE objective with MCC at each node. Let $\mathcal{I}$ be the index set of samples at a node, and assume a constant model $f(x) = c$ for all x in that node. The MCC objective for the node is

$$J_{\mathrm{node}}(c) = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \exp\left(-\frac{(y_i - c)^2}{2\sigma^2} \right), \tag{8}$$

where $\sigma > 0$ is the kernel bandwidth.

The optimal leaf value $c^\star$ is defined as

$$c^\star = \arg\max_{c \in \mathbb{R}} J_{\mathrm{node}}(c). \tag{9}$$

Taking the derivative of (8) w.r.t. c and setting it to zero yields

$$\sum_{i \in \mathcal{I}} \exp\left(-\frac{(y_i - c)^2}{2\sigma^2} \right) (y_i - c) = 0. \tag{10}$$

Rearranging, we obtain

$$\sum_{i \in \mathcal{I}} w_i(c) y_i = c \sum_{i \in \mathcal{I}} w_i(c), \quad w_i(c) = \exp\left(-\frac{(y_i - c)^2}{2\sigma^2} \right). \tag{11}$$

Therefore $c^\star$ satisfies the fixed-point equation

$$c^\star = \frac{\sum_{i \in \mathcal{I}} w_i(c^\star) y_i}{\sum_{i \in \mathcal{I}} w_i(c^\star)}. \tag{12}$$

This equation can be solved iteratively by

$$c^{(t+1)} = \frac{\sum_{i \in \mathcal{I}} w_i^{(t)} y_i}{\sum_{i \in \mathcal{I}} w_i^{(t)}}, \quad w_i^{(t)} = \exp\left(-\frac{(y_i - c^{(t)})^2}{2\sigma^2} \right), \tag{13}$$

initialized, for example, with the sample median of $\{y_i\}_{i \in \mathcal{I}}$. In practice, a small number of iterations (e.g., 3–5) is often sufficient to obtain a robust estimate.

The optimal node score is then

$$J^\star_{\mathrm{node}} = J_{\mathrm{node}}(c^\star). \tag{14}$$

A. Interpretation of the Fixed-Point Update

Unlike the mean (solution of a quadratic loss) or the median (solution of an L_1 loss), the MCC objective in (8) is nonlinear in c and does not admit a closed-form minimizer. Maximizing $J_{\mathrm{node}}(c)$ is equivalent to solving an M-estimation problem with score function

$$\psi_\sigma(e) = \frac{e}{\sigma^2} \exp\left(-\frac{e^2}{2\sigma^2} \right), \tag{15}$$

so that the estimating equation $\sum_{i \in \mathcal{I}} \psi_\sigma(y_i - c) = 0$ is, up to a constant factor, equivalent to the fixed-point condition in (12) [5]. The resulting estimator can be interpreted as a *correntropy-weighted mean*: observations close to c receive weights near one, while the influence of distant observations decays exponentially. In contrast to the mean and median, this yields a redescending influence function in which sufficiently large residuals have essentially zero effect on the estimate.

VI. Robustness Interpretation as an M-Estimator

Maximizing the correntropy objective in (8) is equivalent to minimizing the empirical loss

$$L(c) = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \ell_\sigma(y_i - c), \quad \ell_\sigma(e) = 1 - \exp\left(-\frac{e^2}{2\sigma^2} \right), \tag{16}$$

which defines an M-estimator in the sense of robust statistics [2]. The corresponding score function is

$$\psi_\sigma(e) = \frac{\partial \ell_\sigma(e)}{\partial e} = \frac{e}{\sigma^2} \exp\left(-\frac{e^2}{2\sigma^2} \right). \tag{17}$$

The M-estimation normal equation is

$$\sum_{i\in\mathcal{I}} \psi_\sigma(y_i - c) = 0, \tag{18}$$

which, up to the constant factor $1/\sigma^2$, is equivalent to the stationarity condition derived from (8).

A. IRLS View and Connection to CART

Equation (18) admits an iteratively reweighted least squares (IRLS) interpretation [2]. Writing $e_i^{(t)} = y_i - c^{(t)}$ and

$$w_i^{(t)} = \exp\left(-\frac{\left(e_i^{(t)}\right)^2}{2\sigma^2}\right), \tag{19}$$

a Newton or IRLS step for minimizing (16) leads to the update

$$c^{(t+1)} = \frac{\sum_{i\in\mathcal{I}} w_i^{(t)} y_i}{\sum_{i\in\mathcal{I}} w_i^{(t)}}, \tag{20}$$

which is exactly the fixed-point iteration used in our leaf estimator (Eq. 12). Each step solves a *weighted* least-squares problem with data-dependent weights $w_i^{(t)}$; samples with large residuals receive exponentially small weights and have negligible impact on the updated node constant. In this sense, the MCC leaf estimator can be viewed as a CART leaf estimator in which the simple average is replaced by an IRLS-style, correntropy-weighted average.

B. Influence Function and Redescending Behavior

For scalar location M-estimators, the influence function is proportional to the score [2]. Using (17),

$$\mathrm{IF}_\sigma(e) \propto \psi_\sigma(e) = \frac{e}{\sigma^2}\exp\left(-\frac{e^2}{2\sigma^2}\right). \tag{21}$$

Near the origin, $\psi_\sigma(e) \approx e/\sigma^2$ and the influence grows approximately linearly with e, as in least squares. For large residuals,

$$\lim_{|e|\to\infty} \psi_\sigma(e) = 0, \tag{22}$$

so the influence function is *redescending*: extremely large deviations have vanishing effect on the estimator. This stands in contrast to the unbounded influence of the quadratic loss (CART) and complements the bounded but non-redescending influence of L_1-based LAD trees.

C. Local Quadratic Behavior

A Taylor expansion of the correntropy loss around $e = 0$ gives

$$\ell_\sigma(e) = 1 - \exp\left(-\frac{e^2}{2\sigma^2}\right) = \frac{e^2}{2\sigma^2} + O\left(\frac{e^4}{\sigma^4}\right), \quad |e| \ll \sigma. \tag{23}$$

Thus, for residuals much smaller than the bandwidth σ, MCC behaves like a scaled quadratic loss. In nodes where the residuals are well-controlled the MCC impurity is therefore close to the standard MSE impurity and MCRTree behaves similarly to CART. The main differences arise in regions with large residuals where the exponential term suppresses the contribution of outliers.

D. Non-Convexity and Practical Choice of σ

For very small bandwidths, the loss (16) can become non-convex in c and exhibit multiple local minima [5]. In this regime the IRLS fixed-point iteration may in principle converge to different local solutions depending on initialization. To avoid pathological behavior we restrict attention to moderate bandwidths and initialize $c^{(0)}$ at the sample median in each node. Empirically, this combination produces stable and reproducible estimates in all experiments, while still providing strong robustness gains over pure L_2 optimization.

VII. MCC-Based Node Impurity and Split Criterion

We define the MCC-based node impurity as a monotone transform of the optimal correntropy. In our implementation we use

$$I_{\mathrm{MCC}}(\mathcal{I}) = -\log\left(\varepsilon + J^\star_{\mathrm{node}}\right), \tag{24}$$

where $\varepsilon > 0$ is a small constant, so that larger correntropy corresponds to lower impurity. This log transform improves numerical stability when $J^\star_{\mathrm{node}}$ is close to 1.

For a candidate split s that divides $\mathcal{I}$ into $\mathcal{I}_L$ and $\mathcal{I}_R$, with $n = |\mathcal{I}|$, $n_L = |\mathcal{I}_L|$, and $n_R = |\mathcal{I}_R|$, we compute the optimal leaf values $c_P^\star$, $c_L^\star$, and $c_R^\star$ for parent, left child, and right child, along with their scores $J_P^\star$, $J_L^\star$, and $J_R^\star$. The MCC gain of the split is defined as

$$\Delta I_{\mathrm{MCC}}(s) = I_{\mathrm{MCC}}(\mathcal{I}) - \left(\frac{n_L}{n} I_{\mathrm{MCC}}(\mathcal{I}_L) + \frac{n_R}{n} I_{\mathrm{MCC}}(\mathcal{I}_R)\right). \tag{25}$$

The algorithm chooses the split s that maximizes $\Delta I_{\mathrm{MCC}}(s)$, subject to standard stopping conditions such as minimum node size and maximum tree depth.

VIII. Kernel Bandwidth Selection

The choice of kernel bandwidth σ controls the robustness of the MCC objective. A small σ yields strong suppression of large residuals and more mode-seeking behavior, while a large σ recovers behavior closer to MSE [5], [6].

In this work we use a global bandwidth of the form

$$\sigma = \alpha \cdot \mathrm{MAD}(\{y_i\}_{i=1}^N), \tag{26}$$

where

$$\mathrm{MAD}(y) = \mathrm{median}_i\, |y_i - \mathrm{median}_j y_j| \tag{27}$$

is the median absolute deviation, and $\alpha > 0$ is a scale factor. Using a MAD-based scale makes the bandwidth robust to outliers in the response.

Rather than tuning a separate bandwidth at each node, we treat α as a single hyperparameter per dataset. In the experiments below, α is selected from a small grid on the training set (e.g., $\alpha \in \{0.25, 0.5, \ldots, 3.0\}$ for the heavy-tailed synthetic data and $\alpha \in \{0.5, 1.0, \ldots, 5.0\}$ for Beijing PM2.5) by minimizing validation MAE. A dedicated bandwidth-sensitivity study in Section IX-D illustrates how performance varies across the grid and confirms that MCRTree is robust over a broad range of α values.

Algorithm 1 Maximum Correntropy Regression Tree (MCRTree)

Input: Data $\{(x_i, y_i)\}_{i=1}^{N}$, max depth D, min samples per split, min samples per leaf, bandwidth σ.
Initialize the root node with all indices $\mathcal{I} = \{1, \ldots, N\}$.
Recursively build the tree:
  For a node with index set $\mathcal{I}$ and depth d:
  Compute $c_P^\star$ via fixed-point iterations using σ.
  Compute $J_P^\star$ and $I_{\mathrm{MCC}}(\mathcal{I})$.
  If $d \geq D$ or $|\mathcal{I}|$ below threshold: make the node a leaf with prediction $c_P^\star$ and return.
  For each feature and candidate threshold:
    Split $\mathcal{I}$ into $\mathcal{I}_L$ and $\mathcal{I}_R$.
    If either child violates the min leaf size, skip.
    For each child, compute $c^\star$, $J^\star$, and I_{MCC}.
    Compute the split gain $\Delta I_{\mathrm{MCC}}(s)$.
  Select the split $s^\star$ with maximum gain; if gain is below tolerance, make the node a leaf.
  Otherwise, create left and right children and recurse on each.

Algorithm 1 summarizes the training procedure for the proposed Maximum Correntropy Regression Tree (MCRTree) with a global bandwidth.

IX. Experiments

We evaluate MCRTree on controlled synthetic datasets with either Gaussian or heavy-tailed noise, and on the Beijing PM2.5 air-quality dataset [7], which exhibits strong non-Gaussian variability and sensor-induced spikes.

Our objective is to isolate the effect of replacing the quadratic impurity measure in a *single* regression tree with the Maximum Correntropy Criterion. We compare MCRTree against standard squared-error CART [1] and an LAD tree implemented via scikit-learn's `criterion="absolute_error"`, which performs L_1 splitting with median leaf predictions [2], [14].

The LAD tree provides a principled robust baseline with bounded influence, enabling a meaningful comparison with the redescending behavior induced by MCC.

For all tree-based models we use the same structural hyperparameters: maximum depth 5, a minimum of 20 samples required to split, and at least 5 samples per leaf. For MCRTree, the correntropy bandwidth is defined as $\sigma = \alpha \cdot \mathrm{MAD}(y)$ on the training labels. For each dataset, we first inspect a small grid of α values using held-out splits and then fix a single α for all reported runs (for the final experiments we use $\alpha = 1.5$ for both synthetic problems and $\alpha = 2.5$ for Beijing PM2.5). Each MCRTree leaf value is computed using 5 iterations of the fixed-point update in Eq. 12.

All results are averaged over 10 random train/test splits (70%/30%). We report mean squared error (MSE), mean absolute error (MAE), and median absolute error (MedAE), the latter being particularly informative under heavy-tailed noise. Mean $\pm$ standard deviation are reported for all metrics.

TABLE I
Synthetic regression with Gaussian noise: test-set performance (mean $\pm$ std over 10 splits).

Model	MSE	MAE	MedAE
CART (MSE)	3.204 ± 0.168	1.427 ± 0.034	1.202 ± 0.044
LAD Tree	3.335 ± 0.126	1.457 ± 0.027	1.226 ± 0.044
MCRTree (MCC)	3.254 ± 0.149	1.438 ± 0.033	1.211 ± 0.033

For real-data preprocessing, we report the exact handling of missing targets and categorical covariates in the corresponding dataset subsection.

A. Synthetic Regression: Gaussian Noise

We first consider a linear regression model with Gaussian noise to verify that MCRTree does not degrade performance when the generating process is aligned with the MSE objective. Features are drawn as $x_i \sim \mathcal{N}(0, I_d)$ with $d = 10$, the true weights w are drawn once from $\mathcal{N}(0, I_d)$, and the response is

$$y_i = w^\top x_i + \varepsilon_i, \tag{28}$$

with $\varepsilon_i \sim \mathcal{N}(0, 1)$. We generate $N = 2000$ samples per split.

Table I reports the results (for the current implementation, we used a single MAD-based bandwidth across all splits). As expected, all three trees perform similarly. The MSE tree achieves the lowest MSE; the LAD tree and MCRTree slightly increase MSE but remain very close in MAE and MedAE. This confirms that the correntropy-based objective does not introduce a systematic penalty in the light-tailed regime. This is consistent with the well-known local quadratic behavior of MCC: under light-tailed noise, the correntropy objective behaves similarly to an L_2 criterion, so major deviations from CART are not expected.

B. Synthetic Regression: Heavy-Tailed Noise

We next consider a heavy-tailed setting designed to stress robustness. The feature generation and true weights are as before, but the noise is now a contaminated distribution:

$$\varepsilon_i \sim (1 - \pi)\,\mathcal{N}(0, 1) + \pi\,\mathrm{Laplace}(0, b), \tag{29}$$

with contamination rate $\pi = 0.2$ and outlier scale $b = 10$. This creates frequent, large-magnitude deviations from the underlying linear model.

Table II summarizes the results for the selected bandwidth. Both LAD and MCRTree substantially reduce MAE and MedAE relative to the standard MSE tree, confirming the benefits of robust objectives under heavy-tailed noise. In this configuration, MCRTree slightly outperforms LAD on both MAE and MedAE while remaining competitive in MSE. The gains in MedAE are more pronounced, reflecting the downweighting of extreme residuals by correntropy.

C. Beijing PM2.5 Air Quality

The Beijing PM2.5 dataset [7] contains hourly measurements of fine particulate matter ($\mathrm{PM}_{2.5}$) together with meteorological covariates. We use $\mathrm{PM}_{2.5}$ as the target and meteorological/time variables as inputs, including one-hot encodings

TABLE II
SYNTHETIC REGRESSION WITH HEAVY-TAILED NOISE: TEST-SET PERFORMANCE (MEAN ± STD OVER 10 SPLITS).

Model	MSE	MAE	MedAE
CART (MSE)	41.65 ± 6.32	3.753 ± 0.198	2.121 ± 0.108
LAD Tree	37.03 ± 5.57	3.297 ± 0.171	1.774 ± 0.067
MCRTree (MCC)	37.50 ± 5.69	$\mathbf{3.281 \pm 0.174}$	$\mathbf{1.747 \pm 0.068}$

TABLE III
BEIJING PM2.5: TEST-SET PERFORMANCE (MEAN ± STD OVER 10 SPLITS).

Model	MSE	MAE	MedAE
CART (MSE)	5600 ± 123	53.25 ± 0.56	39.92 ± 0.54
LAD Tree	6386 ± 169	$\mathbf{52.06 \pm 0.50}$	$\mathbf{34.18 \pm 0.50}$
MCRTree (MCC)	6460 ± 162	53.00 ± 0.5594	36.10 ± 0.63

of wind direction when present. Rows with missing $PM_{2.5}$ values were discarded before training; no separate feature-imputation step was used in this implementation. After standardization and train/test splitting, we fit CART, LAD, and MCRTree as in the synthetic experiments.

Table III reports the test-set metrics. The LAD tree achieves the lowest MAE and MedAE, reflecting its strong robustness to large pollution spikes. MCRTree improves substantially over CART in terms of MedAE (from 39.92 to 36.10) and slightly in MAE, but remains worse than LAD across all metrics. This behavior is consistent with the heavy-tailed synthetic setting: correntropy reduces the influence of extreme observations relative to L_2 splitting, but does not fully match an L_1 objective that is directly aligned with the noise distribution.

D. Bandwidth Sensitivity

To better understand the impact of the bandwidth parameter, we perform a sensitivity analysis by varying α in the definition $\sigma = \alpha \cdot \mathrm{MAD}(y)$ and re-training MCRTree for each value. We focus on MAE and MedAE, averaged over the same 10 train–test splits as above.

For the heavy-tailed synthetic dataset, we sweep $\alpha \in \{0.25, 0.5, \ldots, 3.0\}$. The resulting curves are shown in Fig. 1. MAE varies only weakly across the entire range (within roughly 3.26–3.39), while MedAE exhibits a broad minimum around $\alpha \approx 1$–1.5 (note that the implementation used a finer internal grid; the printed values are rounded to one decimal place). Very small bandwidths slightly increase both MAE and MedAE, consistent with an overly aggressive mode-seeking behavior. Very large bandwidths gradually move the estimator toward an MSE-like regime.

For the Beijing PM2.5 dataset, we sweep $\alpha \in \{0.5, 1.0, \ldots, 5.0\}$; see Fig. 2. Here, MAE improves as α decreases from very large values and stabilizes around $\alpha \approx 2$–3, whereas MedAE is smallest for α near 0.5–1.0 and increases monotonically thereafter. This is intuitive: small bandwidths emphasize typical, central behavior and largely ignore pollution spikes, leading to low MedAE but higher MAE. Larger bandwidths retain more sensitivity to spikes

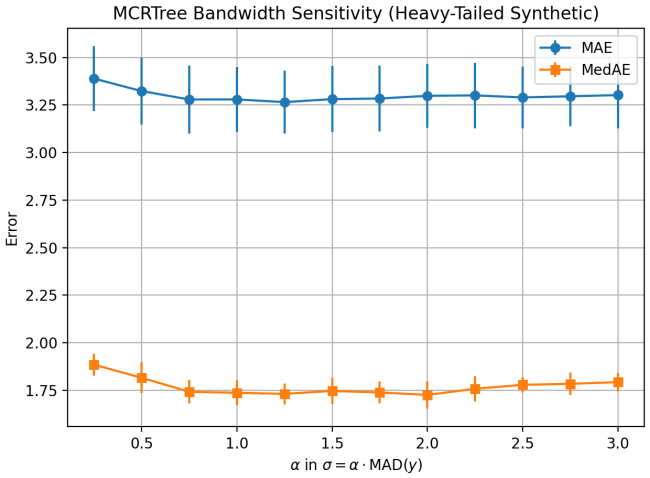

Fig. 1. MCRTree bandwidth sensitivity on the heavy-tailed synthetic dataset. Error bars show ± one standard deviation over 10 splits.

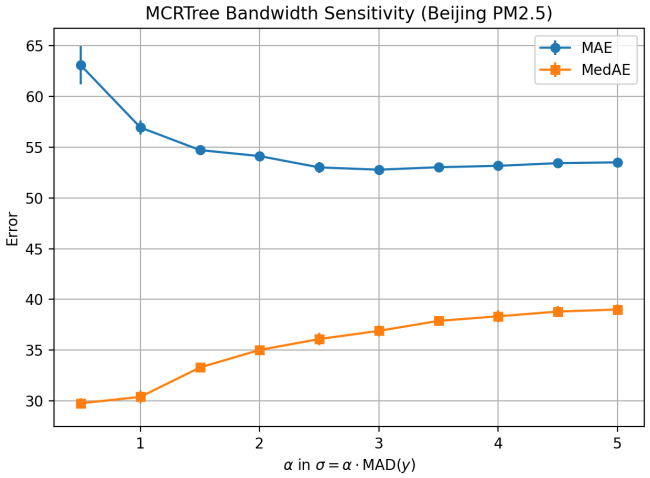

Fig. 2. MCRTree bandwidth sensitivity on the Beijing PM2.5 dataset. Error bars show ± one standard deviation over 10 splits.

and thus favor MAE at the cost of higher median error. Overall, the curves demonstrate that MCRTree is robust across a reasonably wide range of α and that extremely small or large bandwidths behave in line with MCC theory.

E. Discussion

The empirical study is intentionally focused on controlled synthetic settings and one real-world benchmark so that the effect of replacing the node objective can be isolated without introducing confounding changes in model class or feature engineering.

The synthetic experiments highlight two regimes. Under Gaussian noise, MCRTree essentially matches the MSE tree in MAE and MedAE, with only a small increase in MSE, indicating the correntropy objective does not impose a substantial penalty in well-behaved settings. Under heavy-tailed contamination, both LAD and MCRTree substantially reduce

TABLE IV
MEAN TRAINING TIME PER TREE (SECONDS, MEAN OVER 10 SPLITS).

Dataset	CART (MSE)	LAD Tree	MCRTree (MCC)
Synthetic (Gaussian)	0.0081	0.0878	3.35
Synthetic (Heavy-tailed)	0.0080	0.0867	3.13
Beijing PM2.5	0.0450	18.82	13.00

MAE and MedAE relative to the MSE tree, confirming the benefit of robust objectives; in our experiments, MCRTree behaves as a compromise between L_2 and L_1 splitting.

On the Beijing PM2.5 dataset, which contains strong non-Gaussian spikes and heavy-tailed sensor noise, the same pattern appears: LAD regression trees are very effective at shrinking large errors in the tails, while MCRTree offers a middle ground between pure L_2 and pure L_1 behavior. The bandwidth-sensitivity curves further show that this trade-off can be steered by the choice of α: smaller values emphasize median behavior at the cost of higher MAE, while moderate values around $\alpha \approx 2$–3 yield MAE close to (though still worse than) LAD, and clearly improve over pure L_2 splitting.

X. COMPUTATIONAL COMPLEXITY AND LIMITATIONS

MCRTree follows the CART structure but incurs higher split-evaluation cost because MCC leaf fitting and correntropy scoring must be computed for candidate children. At a node with n samples and d features, sorting is performed once per feature and split positions are scanned. Without incremental updates, each candidate split requires recomputing $(c^\star, J^\star)$ via T fixed-point iterations, leading to worst-case $O(Tdn^2)$ complexity per node. In contrast, CART-style MSE (and typical L_1 trees) rely on cumulative statistics with $O(1)$ updates per split in optimized implementations.

Empirically (Table IV), MCRTree is significantly slower than CART and LAD on the synthetic tasks, and remains slower than CART but comparable to LAD on Beijing PM2.5. These differences reflect both algorithmic overhead and implementation maturity (Python vs. optimized C). Efficient implementations would likely require histogram-based or approximate split evaluation.

Current limitations are primarily implementation- and scope-related: the present Python/NumPy prototype does not use incremental or histogram-based split evaluation, bandwidth is global rather than node-adaptive, experiments are limited to two synthetic settings and one real dataset, and only exact axis-aligned splits are considered. Our contribution is methodological; efficient large-scale implementations are future work.

XI. CONCLUSION

We have proposed a regression tree based on the Maximum Correntropy Criterion. By replacing the MSE objective with a correntropy-based criterion at each node, we obtain robust leaf estimators and a split selection mechanism that are less sensitive to outliers & heavy-tailed noise. The resulting tree preserves the interpretability of standard regression trees while inheriting the robustness of MCC-based estimators. Synthetic experiments and results on Beijing PM2.5 confirm that MCRTree behaves similarly to standard trees under Gaussian noise and improves robustness under heavy-tailed noise. The bandwidth-sensitivity analysis clarifies how the kernel size trades off between median/mean performance, and indicates that MCRTree is stable over a broad range of reasonable bandwidths. We believe that correntropy-based objectives provide a useful addition to the toolbox of robust trees.

XII. ACKNOWLEDGMENTS

The authors acknowledge the use of ChatGPT (GPT-5.2 Thinking) by OpenAI [15] to support drafting and language refinement in the Introduction and the Conclusion.

REFERENCES

[1] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Belmont, CA: Wadsworth, 1984.
[2] P. J. Huber, "Robust estimation of a location parameter," *Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, 1964.
[3] S. Dlugosz, *rpart.LAD: Least Absolute Deviation Regression Trees*, 2023, r package version 0.1.3. [Online]. Available: https://cran.r-project.org/package=rpart.LAD
[4] N. Meinshausen, "Quantile regression forests," *Journal of Machine Learning Research*, vol. 7, pp. 983–999, 2006.
[5] W. Liu, P. P. Pokharel, and J. C. Principe, "Correntropy: Properties and applications in non-gaussian signal processing," *IEEE Transactions on Signal Processing*, vol. 55, no. 11, pp. 5286–5298, 2007.
[6] J. C. Principe, *Information Theoretic Learning: Rényi's Entropy and Kernel Perspectives*. New York, NY: Springer, 2010.
[7] S. X. Chen, J. Guo, and X. Wang, "Assessing beijing's $PM_{2.5}$ pollution: Severity, weather impact, APEC and winter heating," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 471, no. 2182, p. 20150257, 2015.
[8] G. Galimberti, M. Pillati, and G. Soffritti, "Robust regression trees based on m-estimators," *Statistica*, vol. 67, no. 2, pp. 173–190, 2007.
[9] S. Basalamah and A. Sihabuddin, "A huber m-estimator algorithm and decision tree regression approach to improve the prediction performance of datasets with outlier," *International Journal of Intelligent Engineering and Systems*, vol. 17, no. 1, pp. 1–9, 2024.
[10] S. C. Tan, "Decision tree regression with residual outlier detection," *Journal of Data Science and Intelligent Systems*, vol. 3, no. 2, pp. 126–135, 2025.
[11] T. Cao, L. Lu, and T. Jiang, "Robust regression in environmental modeling based on bayesian additive regression trees," *Environmental Modeling & Assessment*, vol. 29, pp. 31–43, 2024.
[12] J. Li, Q. Hu, X. Liu, and Y. Zhang, "Augmented maximum correntropy criterion for robust geometric perception," *IEEE Transactions on Robotics*, 2024.
[13] Y. Zheng, S. Wang, and B. Chen, "Generalized multikernel correntropy based broad learning system for robust regression," *Information Sciences*, vol. 678, p. 121026, 2024.
[14] R. Koenker, *Quantile Regression*. Cambridge University Press, 2005.
[15] OpenAI, "ChatGPT (gpt-5.2 thinking)," Large Language Model, 2026, accessed: 2026-01-01. [Online]. Available: https://chat.openai.com/