SAME INFORMATION, DIFFERENT GENERALIZATION: REPRESENTATION SENSITIVITY IN NEURAL NETWORK REGRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Invertible feature transformations preserve predictive information, yet can affect finite-sample learning performance. We study this representation sensitivity in controlled synthetic tabular regression. Our paired experiments compare transformations with the same normalized population error under optimal affine approximation, thereby matching the total amount of induced nonlinearity while varying its structure. Despite this matching, neural networks can exhibit different learning curves, and the ordering of representation penalties can reverse depending on whether the transformed coordinates define the inputs or the target relationship. In contrast, XGBoost is nearly unaffected by the transformations. We characterize these differences by decomposing nonlinear target energy across polynomial degrees using projections adapted to the observed input distribution. Our results show that neural-network representation sensitivity depends on the target's nonlinear structure in the observed coordinates, the input distribution, and the learner.

1 INTRODUCTION

In supervised learning problems, an invertible transformation of the input features X preserves the information available for prediction, but may not preserve the difficulty of learning the resulting input-output relationship from finite data. If X is replaced by an invertible representation $T(X)$, the Bayes-optimal predictor can be expressed in either coordinate system. However, a finite-sample learning algorithm is affected by the geometry and distribution of its inputs, the complexity of the target function in those coordinates, and the inductive bias of the learning algorithm. Consequently, two statistically equivalent representations can yield substantially different generalization behavior.

A simple example shows how this can happen. Let Z and $U = T(Z)$, where T is an invertible transformation. If the noiseless response is a linear function of U, a learner observing U faces a linear target, while a learner observing Z must fit the forward map T. But if the noiseless response is a linear function of Z, a learner observing Z faces a linear target, while a learner observing U must fit the inverse map T^{-1}. In all cases, changing the observed representation does not remove any predictive information. But it does change the target function in the observed coordinates, as well as the distribution of those coordinates. A clear consequence is that if T is nonlinear, then a linear regression function would not perform equally well in all of these scenarios.

This issue is particularly relevant for tabular learning, where transformations of numerical features are common. Gorishniy et al. (2021) treated data preprocessing as an important component of tabular neural-network performance, while also reporting a dataset for which it was detrimental. Numerical feature embeddings can also improve the performance of tabular neural networks, including MLP-like models (Gorishniy et al., 2022). Holzmüller et al. (2024) compared preprocessing choices for a tuned MLP, including standardizations and normal-quantile transformations. More broadly, large-scale tabular benchmarks have demonstrated systematic differences between neural networks and tree-based learners. Grinsztajn et al. (2022) identified several inductive biases that help explain the strength of tree ensembles on tabular data, including robustness to uninformative features and an ability to fit irregular target functions. McElfresh et al. (2023) across 176 datasets further found that gradient-boosted trees are substantially better suited than neural networks to datasets with skewed or heavy-tailed feature distributions and related dataset irregularities. These findings motivate studying

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

representation changes, but do not directly measure what happens when the same target is learned from two invertibly related sets of coordinates.

One possibility is that performance depends on how much the target departs from a linear function of the observed features. For a standard Gaussian feature Z, we quantify this as the aggregate departure of $T(Z)$ from its best affine approximation using

$$\eta(T) = 1 - \mathrm{Corr}(Z, T(Z))^2,$$

which is the fraction of variance in $T(Z)$ unexplained by the optimal affine predictor based on Z. After normalizing the target variance, the optimal affine residual has the same value when predicting Z from $T(Z)$. Thus, $\eta(T)$ is identical for the forward and inverse problems, even though their target functions and observed input distributions differ. This allows us to ask: do transformations with the same total departure from linearity have the same learning difficulty?

Prior work suggests they may not. Neural networks can exhibit spectral bias, preferentially learning simpler or lower-frequency components of a target function (Rahaman et al., 2019). In a neural tangent kernel analysis, the rate at which a frequency component is learned also depends on input density (Basri et al., 2020). For kernel regression and infinite-width neural networks, Canatar et al. (2021) show that generalization depends on the kernel eigenspectrum together with the alignment of the target function to those eigenfunctions. In high-dimensional linearized models, the polynomial degrees that can be learned depend on sample size (Ghorbani et al., 2021). Karkada et al. (2026) connect polynomial decompositions of the target to kernel learning curves and report related degree-wise behavior in neural networks. Motivated by this work, we examine how the nonlinear part of a target is distributed across polynomial degrees under the observed input distribution. For Gaussian inputs this gives a Hermite decomposition. For transformed inputs, the orthogonal polynomials instead adapt to their generally non-Gaussian distribution.

We study these questions in controlled tabular regression experiments. We generate independent Gaussian features, apply invertible coordinatewise transformations, and form a noisy response from a sum of active features in either the original or transformed representation. We consider four primary settings: 1) matched conditions in which both the features and response have been transformed. In this case, the underlying regression function is linear. 2) Matched conditions when neither the feature nor the response has been transformed. 3) Mismatched conditions in which the learner must approximate the forward map T or 4) its inverse T^{-1}. We calibrate multiple transformation families to have common values of η, then compare learning curves across families, sample sizes, map directions, and learners. We also test whether summaries of degree-wise target energy improve predictions of representation penalties on a held-out transformation family.

Our main contributions are: **1)** We introduce a paired tabular regression design that compares invertibly related observed representations while holding the response fixed within each comparison. It includes forward and inverse mismatches and matches transformation families by their population departure from linearity, η. **2)** We show that equal η does not imply equal neural-network learning difficulty. Representation penalties vary across transformation families and sample sizes, and their ordering can reverse between map directions. **3)** We show that Gaussianizing preprocessing does not remove these penalties but moves them: a quantile transform to Gaussian marginals eliminates the inverse penalty but imposes the forward penalty when the response is linear in the observed features, so whether it helps depends on a relationship that is not observed in practice. **4)** We decompose the noiseless target into orthogonal polynomial components under the observed input distribution. This extends the Hermite analysis for Gaussian inputs to the non-Gaussian inverse setting and distinguishes quadratic from higher-order nonlinear energy. **5)** We test these energy summaries on a held-out transformation family and compare neural networks with tree-based learners. The held-out prediction improves most in the forward direction, while the learner comparison shows that the penalties depend on the learner despite the transformations preserving predictive information.

2 METHODS

2.1 EXPERIMENTAL SETUP

We study how the relationship between observed features and the representation generating the response affects predictive performance. The central experiment uses $p = 50$ features, $a =$

Table 1: Representation conditions for our experiments. The first letter denotes the observed representation and the second the signal representation. B denotes base coordinates and U transformed coordinates.

Condition	Observed predictors	Signal coordinates	Description
BB	B	B	Base matched
BT	B	U	Forward mismatch
TB	U	B	Inverse mismatch
TT	U	U	Transformed matched

10 active features, response-noise standard deviation $\sigma_\varepsilon = 1$, and total sample sizes of $n \in \{250, 500, 1000, 2000, 5000, 10000\}$. Each experimental condition is evaluated over 25 repetitions unless otherwise noted. An outer 80/20 train/test partition separates model fitting and hyperparameter selection from evaluation.

Let $Z = (Z_1, \ldots, Z_p)$ have independent standard normal coordinates. For a coordinate-wise transformation T, define the population-standardized base and transformed representations by

$$B_j = Z_j, \qquad U_j = \frac{T(Z_j) - \mathbb{E}[T(Z_j)]}{\sqrt{\mathrm{Var}(T(Z_j))}}.$$

These population definitions provide the reference model. The empirical standardization used in the finite-sample experiments is described in Section 2.4.

We distinguish the *observed representation*, supplied as input to the learner, from the *signal representation*, whose active coordinates generate the response. For signal representation $S \in \{B, U\}$, the response is a sum of the a active features plus independent noise:

$$Y = \sum_{j=1}^{a} S_j + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2).$$

Table 1 describes the four different settings of our experiments. The first letter in the "Condition" column denotes the observed representation and the second denotes the signal representation. T in these labels denotes the transformed and standardized representation U. In the matched conditions (BB and TT), the regression function is linear in the active observed coordinates. In the forward mismatch condition (BT), it involves the forward transformation. In the inverse mismatch condition (TB), it involves the inverse transformation, accounting for standardization. Because the transformations are invertible on their images, changing the observed representation preserves the information available about the response. Performance differences therefore reflect the behavior of the fitted learning pipelines under different representations.

We measure the effect of a representation mismatch by paired differences in test R^2 between each mismatched condition and the matched condition with the same signal:

$$\Delta_{\text{forward}} = R^2_{TT} - R^2_{BT}, \qquad \Delta_{\text{inverse}} = R^2_{BB} - R^2_{TB}$$

Positive values indicate the mismatched condition performed worse. Under the population reference model, the Bayes-optimal population R^2 is $10/11 \approx 0.909$, since the signal variance is 10 and the independent noise variance is 1.

2.2 Transform families and nonlinearity matching

We consider four monotone transformation families:

$$T_{\text{cubic},\alpha}(z) = z + \alpha z^3, \qquad T_{\text{quintic},\alpha}(z) = z + \alpha z^5,$$

$$T_{\exp,\alpha}(z) = \frac{e^{\alpha z} - 1}{\alpha}, \qquad T_{\text{qc},\alpha}(z) = z + \alpha z^2 + \frac{\alpha^2}{3} z^3,$$

where $\alpha \geq 0$ controls transformation strength and $T_{\exp,0}(z) = z$ is defined by continuity.

We refer to the final family as the quadratic–cubic transformation. Its derivative is $T'_{\mathrm{qc},\alpha}(z) = (1 + \alpha z)^2$. For $\alpha > 0$, it is strictly increasing despite its derivative vanishing at $z = -1/\alpha$. Its inverse does not have a finite derivative at the corresponding transformed value.

For $Z \sim \mathcal{N}(0, 1)$, we define population nonlinearity as

$$\eta(T) = 1 - \mathrm{Corr}(Z, T(Z))^2 = \frac{\min_{b_0, b_1 \in \mathbb{R}} \mathbb{E}\left[\left(T(Z) - b_0 - b_1 Z\right)^2\right]}{\mathrm{Var}(T(Z))}.$$

Thus, $\eta(T)$ is the fraction of transformed variance unexplained by the best affine approximation in the base coordinate. It is invariant to nondegenerate affine rescaling of either variable.

For each family, transformation strengths are numerically calibrated to target $\eta \in \{0.05, 0.10, 0.15, 0.20, 0.30\}$. The absolute discrepancies between target and achieved population nonlinearity are less than 10^{-4}. This matching controls the aggregate departure from an affine relationship while allowing transformation families to differ in spectral allocation, smoothness, and induced feature distributions.

2.3 Spectral decomposition and higher-order energy

We characterize nonlinear signal components through their spectral energy relative to the observed input distribution. Let $X \sim \mu$ where μ denotes the distribution of the observed scalar input, and let $f(X)$ be a scalar signal component standardized to have mean zero and unit variance. For polynomials $\{\phi_k\}_{k=0}^K$ orthonormal with respect to μ, ordered by degree and with $\phi_0 = 1$, define

$$c_k = \mathbb{E}[f(X)\phi_k(X)], \qquad \mathbb{E}[\phi_j(X)\phi_k(X)] = \delta_{jk}.$$

The energy at order k is c_k^2. These energies depend jointly on the signal function, the input distribution, and the chosen polynomial basis.

The degree-K orthogonal projection and its residual energy are

$$f_K(x) = \sum_{k=0}^{K} c_k \phi_k(x), \qquad R_K = \mathbb{E}\left[(f(X) - f_K(X))^2\right].$$

Since the signal is centered and standardized,

$$c_0 = 0, \qquad \sum_{k=1}^{K} c_k^2 + R_K = 1.$$

This finite-projection identity does not require completeness of the polynomial basis.

Forward spectrum. In the forward mismatch condition, the observed scalar input is $Z \sim \mathcal{N}(0, 1)$ and the standardized signal function is

$$f_{\mathrm{F}}(z) = \widetilde{T}(z) = \frac{T(z) - \mathbb{E}[T(Z)]}{\sqrt{\mathrm{Var}(T(Z))}}.$$

We use the orthonormal Hermite polynomials $\{\psi_k\}$ under the standard Gaussian measure, with $\psi_0(z) = 1$ and $\psi_1(z) = z$. The forward coefficients are

$$c_{k,\mathrm{F}} = \mathbb{E}[\widetilde{T}(Z)\psi_k(Z)].$$

Hermite energy is therefore a special case of spectral energy, corresponding to Gaussian inputs.

Inverse spectrum. In the inverse mismatch condition, the observed scalar input is $U = \widetilde{T}(Z)$ and the standardized signal function is

$$f_{\mathrm{I}}(u) = \widetilde{T}^{-1}(u).$$

Let $\{\phi_{k,\mathrm{I}}\}$ be polynomials orthonormal with respect to the distribution of U. The inverse coefficients are

$$c_{k,\mathrm{I}} = \mathbb{E}[f_{\mathrm{I}}(U)\phi_{k,\mathrm{I}}(U)] = \mathbb{E}[Z\phi_{k,\mathrm{I}}(U)].$$

These coefficients describe the inverse signal under its observed input distribution and generally differ from the forward Hermite coefficients.

Aggregate nonlinearity and spectral allocation. Because both Z and U are standardized, their degree-one orthonormal polynomials are z and u, respectively. Hence

$$c_{1,\mathrm{F}} = c_{1,\mathrm{I}} = \mathbb{E}[ZU] = \mathrm{Corr}(Z, T(Z)).$$

For either direction $d \in \{\mathrm{F}, \mathrm{I}\}$ and $K \geq 1$,

$$\sum_{k=2}^{K} c_{k,d}^2 + R_{K,d} = 1 - c_{1,d}^2 = \eta(T).$$

Thus, forward and inverse conditions have the same total nonlinear residual fraction, but may allocate this energy differently across polynomial orders and the projection residual.

For the complete Gaussian Hermite basis, this gives

$$\sum_{k=2}^{\infty} c_{k,\mathrm{F}}^2 = \eta(T).$$

For the inverse polynomial expansion, we retain the finite-order projection residual explicitly rather than assuming completeness.

To summarize spectral allocation, we define the retained higher-order energy as

$$E_{2,d} = c_{2,d}^2, E_{\geq 3,d} = \sum_{k=3}^{K} c_{k,d}^2 + R_{K,d} = \eta - E_{2,d}$$

Unlike the total nonlinear residual fraction $\eta(T)$, its allocation between E_2 and $E_{\geq 3}$ can differ between directions and across transform families with matched $\eta(T)$. We therefore distinguish aggregate nonlinearity from its spectral allocation and use the term *Hermite energy* specifically for the Gaussian-input expansion.

2.4 Finite-sample construction

Base and transformed coordinates are standardized using empirical moments over each generated sample before responses are constructed and the outer train/test split is applied. Because responses therefore depend on jointly estimated sample moments, the finite-sample construction is not exactly i.i.d. To test for the possible effects of data leakage, we ran a control experiment where standardization was performed using population-level statistics, which gave similar results (see Appendix C.1).

Learners. The default learner for our experiments is a ReLU MLP with training-fitted feature standardization. We additionally evaluate a residual MLP inspired by Gorishniy et al. (2021), XGBoost (Chen & Guestrin, 2016) as a monotone-split sanity check, and the pretrained TabICLv2 regressor (Qu et al., 2026). MLP and XGBoost hyperparameters are selected by five-fold cross-validation. Full specifications for the models and the quantile-preprocessing variant are given in Appendix A.1.

2.5 Allocation of Degradation

To test whether regressor performance degradation depends on how nonlinearity is allocated rather than only on its total, we fit four linear regression models of ΔR^2 on $\log n$ and candidate summaries, each with a $\log n$ interaction:

$$\begin{aligned}
\text{Nonlinearity: } & \beta_0 + \beta_1 \log n + \beta_2 \eta + \beta_3 \eta \log n \\
E_2 + E_3 : & \beta_0 + \beta_1 \log n + \beta_2 E_2 + \beta_3 E_2 \log n + \beta_4 E_3 + \beta_5 E_3 \log n \\
E_2 + E_{\geq 3} : & \beta_0 + \beta_1 \log n + \beta_2 E_2 + \beta_3 E_2 \log n + \beta_4 E_{\geq 3} + \beta_5 E_{\geq 3} \log n \\
\text{Curvature: } & \beta_0 + \beta_1 \log n + \beta_2 \eta + \beta_3 \eta \log n + \beta_4 \bar{\kappa} + \beta_5 \bar{\kappa} \log n,
\end{aligned}$$

where $\bar{\kappa} = \mathbb{E}_Z[\kappa(Z)]$, $\kappa(z) = |\tilde{T}''(z)|/(1 + \tilde{T}'(z)^2)^{3/2}$, and $\tilde{T}$ is T standardized to zero mean and unit variance under $Z \sim \mathcal{N}(0, 1)$.

Since $E_{\geq 3} = \eta - E_2$, the $E_2 + E_{\geq 3}$ model is a reparameterization of η plus quadratic energy: its predictors span the same space as $\{\eta, E_2\}$ and their $\log n$ interactions, so it gives identical fitted

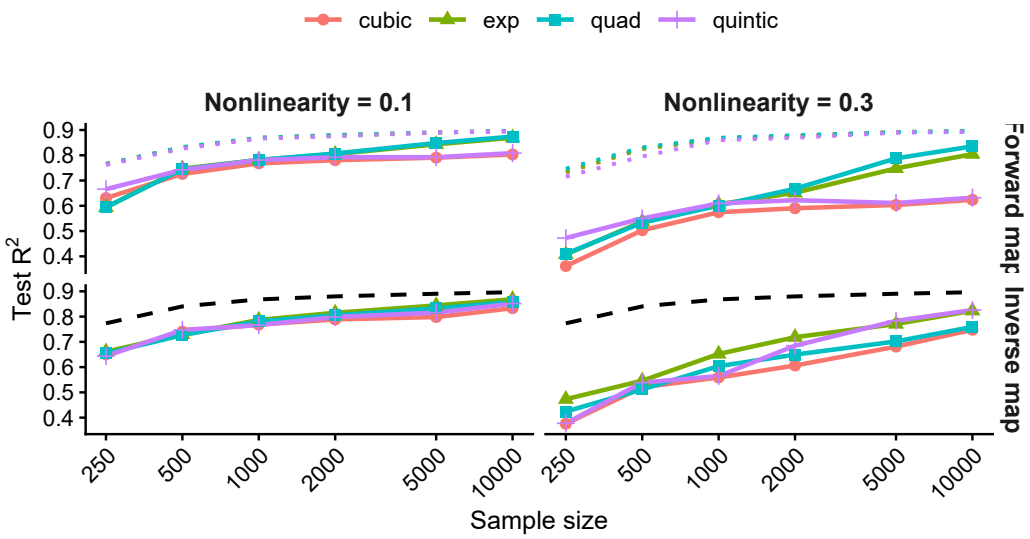

Figure 1: Test R^2 across sample size for transformations matched at $\eta = 0.10$ and $\eta = 0.30$. Solid curves show mismatched conditions: BT for the forward map and TB for the inverse map. In the forward panels, transform-specific dotted curves show the corresponding TT matched controls; in the inverse panels, the black dashed curve shows the common BB control. Although transformations within each panel have the same population nonlinearity η, their finite-sample learning curves differ across transformation families and map directions.

values. Comparing it with the Nonlinearity model therefore asks whether knowing how much of the nonlinearity is quadratic adds information beyond its total.

Held-out evaluation. We fit the four penalty-prediction models using the cubic, quintic, exponential, and quadratic–cubic families. We then evaluate their frozen coefficients on the Yeo–Johnson family (Yeo & Johnson, 2000), which was excluded from fitting and is defined by

$$T_{yeo,a}(Z) = \begin{cases} \frac{(Z+1)^a - 1}{a} \text{ if } a \neq 0 \text{ and } Z \geq 0 \\ log(Z+1) \text{ if } a = 0 \text{ and } Z \geq 0 \\ -\frac{(-Z+1)^{2-a} - 1}{2-a} \text{ if } a \neq 2 \text{ and } Z < 0 \\ -log(-Z+1) \text{ if } a = 2 \text{ and } Z < 0. \end{cases}$$

2.6 Additional experiments

Appendix B–D report controls for alternative learners, optimization and capacity, standardization, transformed active/inactive features, and error localization.

3 Results

3.1 Matched nonlinearity does not match learning curves

Our results show that representation mismatch reduces MLP test performance, and the penalties are generally larger at higher matched values of η, as shown in Figure 1. However, transformations calibrated to the same η can produce substantially different learning curves, showing that aggregate nonlinearity, as measured by η, does not solely determine finite-sample performance difficulty. For example, at $\eta = 0.3$, the cubic and quintic transformations are among the most difficult in the forward condition, while the quintic transformation is among the easiest in the inverse condition at larger sample sizes. The relative ordering of transformation families therefore depends on which representation is observed.

Table 2: R^2 for each of the models described in Section 2.5 on the held-out Yeo-Johnson family.

Model	Forward Map	Inverse Map
Nonlinearity	0.826	0.983
$E_2 + E_3$	0.984	0.866
$E_2 + E_{\geq 3}$	0.985	0.987
Curvature	0.826	0.980

The transformed matched controls also vary across families. We therefore assess each mismatch against the matched condition with the same response, using the paired penalties Δ_{forward} and Δ_{inverse}. Matching η captures a broad relationship between departure from linearity and performance loss, but it does not determine the size or sample-size trajectory of these penalties. Results at the remaining values of η and plots with uncertainty intervals appear in Appendices B.1 and B.4, respectively. Generally, higher values of η lead to larger penalties, especially at smaller sample sizes.

Figure 1 also suggests a preprocessing implication. Mapping transformed inputs back toward Gaussian coordinates moves TB toward its BB control when the signal is defined in base coordinates, but moves TT toward the forward mismatch BT when the signal is defined in transformed coordinates. Section 3.4 tests this implication using quantile preprocessing.

3.2 Polynomial allocation and prediction on a held-out family

Although η is the same for the forward and inverse maps, their nonlinear residuals can have different polynomial decompositions under the respective observed input distributions. The difference is visible even within a transformation family. At $\eta = 0.30$, the fraction of nonlinear variance not captured by the degree-three projection is 0.167 for the forward quintic map and 0.927 for its inverse. For the cubic map, the corresponding fractions are approximately zero and 0.833 (Appendix A.3). Thus, the forward decomposition does not determine how the inverse target is represented under its input distribution.

We test whether this allocation helps predict representation penalties on the held-out Yeo–Johnson family. The four regression models in Section 2.5 were fitted on the original transformation families, and their coefficients were fixed before evaluating Yeo–Johnson outcomes. Table 2 reports the R^2 for predicting the penalty ΔR^2. The model using η alone obtains $R^2 = 0.826$ in the forward direction and 0.983 in the inverse direction. Separating quadratic energy E_2 from the remainder $\eta - E_2$ raises these values to 0.985 and 0.987, respectively. The improvement is therefore substantial for the forward map and modest for the inverse map, where η alone already predicts well.

The model using only E_2 and E_3 performs similarly in the forward direction ($R^2 = 0.984$) but less well in the inverse direction ($R^2 = 0.866$). The curvature model provides nearly identical results to the model using η alone (Table 2 and Figure 2). Thus, adding the tested curvature summary provides little additional predictive benefit beyond η and its interaction with sample size on this held-out family. Together, these comparisons provide evidence that the split between quadratic and remaining nonlinear variation helps predict penalties for this held-out family, particularly in the forward direction.

3.3 Learner dependence and robustness checks

The performance penalties are much smaller for some learners. XGBoost's paired representation penalties remain within approximately 4×10^{-4} of zero in both directions (Appendix C.3). This is expected for tree-based learners as monotone transformations preserve the ordering of each feature, and hence the partitions available to axis-aligned tree splits. In contrast, the performance of a ResNet tabular model degrades across all transformation families as η increases for $n = 500$, similar to the default neural network model (Appendix D.1). TabICLv2 (Qu et al., 2026) at $n = 2000$ is less sensitive to changes in η, especially in the inverse condition (Appendix D.2). These comparisons show that representation sensitivity depends on both the learner and sample size.

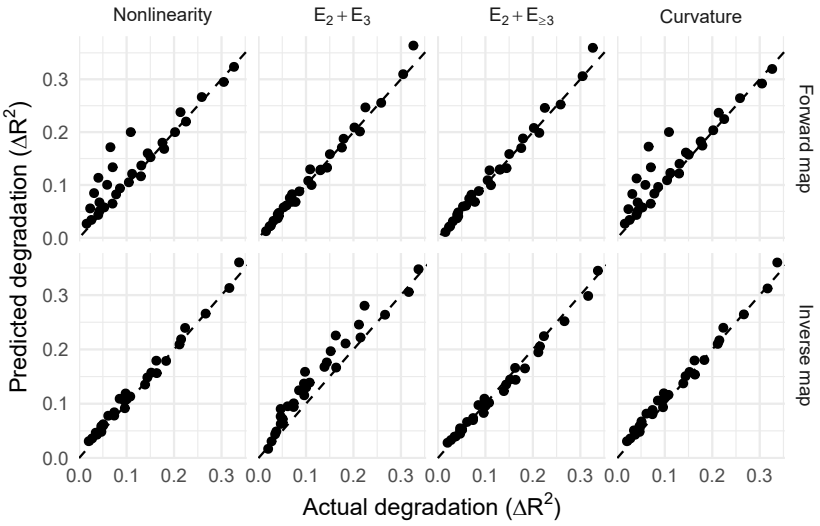

Figure 2: Predicted versus observed ΔR^2 for the held-out Yeo-Johnson transformation family. Regression coefficients were fit using only the four development transformation families and frozen before Yeo-Johnson degradation outcomes were evaluated. Columns compare the Nonlinearity model, degree-specific energy model, higher-order energy model, and curvature model; rows show forward and inverse mappings. The dashed line denotes perfect prediction.

Two controls address whether the primary MLP result is specific to its training and data construction. The performance penalties persist when the training-iteration budget is increased at $n = 500$ and when population moments replace sample-specific moments for standardization (Appendix C.1–C.2). Additional analyses show negligible effects from transforming inactive features and positive excess error within the central region of the signal distribution, as well as in its tails (Appendix B.2–B.3).

3.4 Gaussianizing inputs can help or hurt

Because each transformation T is monotonic, Z and $T(Z)$ have the same ranking of data points. A quantile transform to normal marginals therefore maps their paired samples to nearly the same observed features. This provides a direct way to examine whether Gaussianizing inputs improves performance regardless of which representation generates the response.

Our experiments show it does not. Figure 3 shows that at $\eta = 0.3$, normal-quantile preprocessing improves test R^2 in the inverse mismatch condition (TB) by 0.204 at $n = 2000$ and 0.110 at $n = 10,000$, relative to the standard-scaled MLP baseline. In the transformed matched condition (TT), where the response is linear in the observed coordinates before preprocessing, it instead reduces test R^2 by 0.232 and 0.170, respectively. Mapping to uniform marginals generally performs worse than mapping to normal marginals. Gaussianizing the features therefore favors the base signal representation in this experiment. Its effect depends on how the response relates to the coordinates supplied to the learner.

4 Conclusion and Limitations

Our paired experiments show that invertible changes of observed features can change finite-sample neural network performance without removing predictive information. Transformation families matched on their population departure from the best affine predictor, η, produce different learning curves, and their relative difficulty can reverse between the forward and inverse maps. In general, for a standard neural network, higher levels of nonlinearity (η) result in reduced learner performance in terms of test R^2.

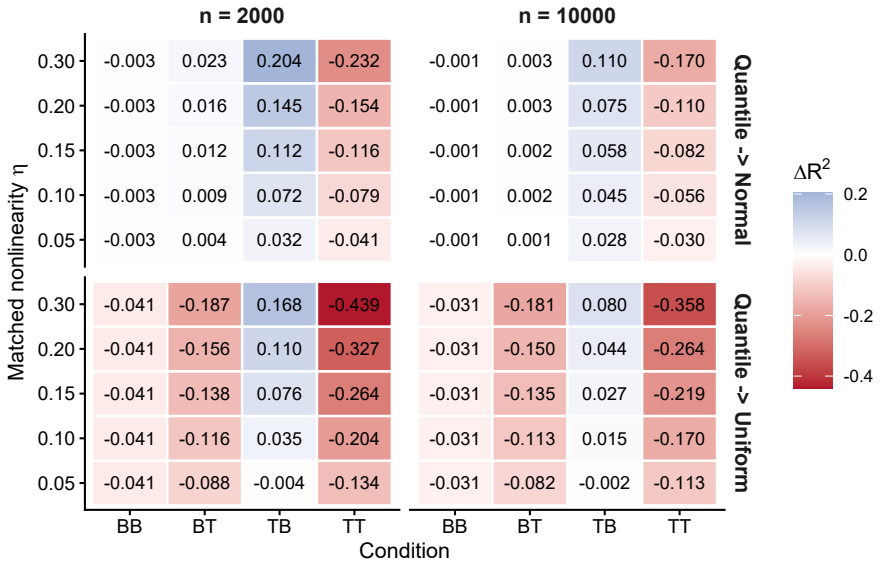

Figure 3: Effect of quantile preprocessing on predictive performance relative to the standard-scaled MLP baseline across the four representation conditions. Each cell shows the mean difference $R^2_{\text{quantile}} - R^2_{\text{baseline}}$ over the first 10 paired repetitions, averaged across transformation families. Positive values indicate improved test performance after quantile preprocessing, while negative values indicate reduced performance. Rows compare quantile mappings to normal and uniform marginals, and columns show results at $n = 2{,}000$ and $n = 10{,}000$.

In addition, we find that while this measure of nonlinearity is informative, it does not fully determine representation sensitivity. Decomposing the target under the observed input distribution provides a more detailed description of its nonlinear structure. Separating quadratic energy from the remaining nonlinear residual improves prediction of performance penalties on the held-out Yeo-Johnson family in the forward direction. In contrast, the improvement is very small in the inverse direction, where η alone predicts well. Sensitivity also varies by learner. A ResNet-style neural network shows similar qualitative behavior at $n = 500$, while TabICLv2 is more robust at $n = 2000$. XGBoost is nearly invariant to the representation transformations. Quantile preprocessing illustrates a practical consequence. Finally, Gaussianizing inputs can improve performance when the signal is linear in the base coordinates and reduce it when the signal is linear in the transformed coordinates.

Several limitations exist. Our findings come from a controlled synthetic setting with independent Gaussian latent features, an additive response, and coordinatewise monotone transformations. They do not establish how often the same effects arise in real tabular data, where predictors may be correlated, targets may involve interactions, noise may be heterogeneous, and latent representations are unknown. The penalty-prediction models were fitted on four transformation families and evaluated on one held-out family, so their predictive value across a broader range of transformations remains uncertain. Most learning curves use one MLP pipeline, while the other neural learners were evaluated at selected sample sizes. The main data construction also standardizes each generated sample before the train/test split. A population-standardization control preserves the qualitative effect, but other data construction and preprocessing choices may yield different results. Testing additional transformations and real tabular datasets would clarify the scope of these findings, and is left for future work. Nevertheless, within the setting studied here, representation sensitivity reflects the target's structure in the observed coordinates, the input distribution, and the learner's inductive bias.

REPRODUCIBILITY

We took great care to ensure the reproducibility of the experiments. This includes creating a repository containing all of the experiments, the exact packages used when the Python scripts

were run, and the results that were achieved when running each file. The data was generated and machine learning models trained using Python. The analysis of results and the production of the figures were performed in R, and a renv lockfile is provided in the repository as well. The higher-order energy calculations and curvature calculations were done using the Jupyter notebooks found in the repository. The link is https://anonymous.4open.science/r/tabular-transform-research-7F4E

AI USE DISCLOSURE

LLMs were used for certain portions of the project. These use cases include assisting in code generation, finding relevant sources in the literature, asking for constructive criticism on the research methodology and experiments, and for assisting in drafting parts of the paper. Actions taken to ensure the validity of these claims were reviewing the code and ensuring it does what was intended to do, ensuring sources cited contain accurate citing information as well as ensuring they contain the relevant information, and carefully reading and directly revising any AI-generated text to ensure accuracy. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI. LLMs were also used to check mathematical derivations and reported values against the experimental outputs, to write code that produced a summary table from existing results, and to suggest edits to the manuscript text; all such output was verified against the underlying data and code by the authors.

REFERENCES

Ronen Basri, Meirav Galun, Amnon Geifman, David Jacobs, Yoni Kasten, and Shira Kritchman. Frequency bias in neural networks for input of non-uniform density. In *International conference on machine learning*, pp. 685–694. PMLR, 2020.

Abdulkadir Canatar, Blake Bordelon, and Cengiz Pehlevan. Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks. *Nature Communications*, 12(1):2914, 2021. doi: 10.1038/s41467-021-23103-1. URL https://www.nature.com/articles/s41467-021-23103-1.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794. ACM, 2016. doi: 10.1145/2939672.2939785.

Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Linearized two-layers neural networks in high dimension. *Annals of Statistics*, 49(2):1029–1054, 2021.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. In *Advances in Neural Information Processing Systems*, volume 34, pp. 18932–18943, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/9d86d83f925f2149e9edb0ac3b49229c-Abstract.html.

Yury Gorishniy, Ivan Rubachev, and Artem Babenko. On embeddings for numerical features in tabular deep learning. *Advances in Neural Information Processing Systems*, 35:24991–25004, 2022.

Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-0037. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/0378c7692da36807bdec87ab043cdadc-Abstract-Datasets_and_Benchmarks.html.

David Holzmüller, Léo Grinsztajn, and Ingo Steinwart. Better by default: Strong pre-tuned mlps and boosted trees on tabular data. In *Neural Information Processing Systems*, 2024.

Dhruva Karkada, Joseph Turnbull, Yuxi Liu, and James B. Simon. Predicting kernel regression learning curves from only raw data statistics. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=nn5Vf6GEsV.

Duncan C. McElfresh, Sujay Khandagale, Jonathan Valverde, Vishak Prasad C., Ganesh Ramakrishnan, Micah Goldblum, and Colin White. When do neural nets outperform boosted trees on tabular data? In *Advances in Neural Information Processing Systems*, volume 36, pp. 76336–76369, 2023. doi: 10.52202/075280-3337. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/f06d5ebd4ff40b40dd97e30cee632123-Abstract.html.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICLv2: A better, faster, scalable, and open tabular foundation model. *arXiv preprint arXiv:2602.11139*, 2026. URL https://arxiv.org/abs/2602.11139.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5301–5310. PMLR, 2019. URL https://proceedings.mlr.press/v97/rahaman19a.html.

In-Kwon Yeo and Richard A. Johnson. A new family of power transformations to improve normality or symmetry. *Biometrika*, 87(4):954–959, 2000. doi: 10.1093/biomet/87.4.954.

A Implementation Details

A.1 Learner Specifications

The learner implementation table summarizes the model architectures, key hyperparameters, optimization procedures, and stopping criteria used for each predictive method. These settings define the implementations used throughout the experiments and are held fixed across transformation conditions unless otherwise noted.

Table 3: Learner configurations. LogU denotes log-uniform sampling.

Learner	Specification
MLP	scikit-learn `MLPRegressor`, ReLU, Adam. Pipeline: `StandardScaler` $\rightarrow$ MLP, fit on training folds only. Search space: hidden layers $\in \{(32), (64), (32, 16), (64, 32)\}$; L_2 penalty $\alpha \sim \mathrm{LogU}(10^{-5}, 10^{-2})$; initial learning rate $\sim \mathrm{LogU}(10^{-4}, 10^{-2})$; batch size $\in \{32, 64, 128\}$. Early stopping (10% validation split, patience 15), at most 500 epochs.
MLP (quantile)	Identical, with `StandardScaler` replaced by `QuantileTransformer` (1000 quantiles, output distribution normal or uniform), fit on training folds only.
ResNet	Residual MLP inspired by Gorishniy et al. (2021), implemented in PyTorch: linear input layer ($50 \rightarrow 64$) with ReLU, three residual blocks (Linear–ReLU–Dropout–Linear, skip connection, then ReLU; width 64, dropout 0.05, no normalization layers), and a linear output head (28,289 parameters). Fixed configuration without hyperparameter search: AdamW, learning rate 10^{-3}, weight decay 10^{-4}, batch size 64, MSE loss, at most 300 epochs. Early stopping on a 20% validation split of the training data (patience 30), restoring the best weights. Standardization fit on the remaining 80% of the training data.
XGBoost	`XGBRegressor`, squared error, histogram tree method. Search space: number of trees $\sim \mathcal{U}\{10, \ldots, 1000\}$; learning rate $\sim \mathrm{LogU}(10^{-3}, 10^{-1})$; max depth $\sim \mathcal{U}\{1, \ldots, 5\}$.
TabICLv2	Released pretrained checkpoint `tabicl-regressor-v2-20260212.ckpt` (Qu et al., 2026), used as a regressor with default settings. No task-specific training or hyperparameter search; predictions are made in context from the training rows. Evaluated at $n = 2000$.
All	The MLP and XGBoost use random search over 10 configurations, scored by 5-fold CV mean squared error (`KFold`, shuffled, fixed seed) within the outer 80% training split; the best configuration is refit on the full training split. The ResNet uses the fixed configuration above. All learners are evaluated on the same 20% test split and see identical data, splits and noise within a replication.

A.2 Computing Spectral and Curvature Summaries

Quadrature. Let $\{(z_i, w_i)\}_{i=1}^m$ be the $m = 200$ probabilists' Gauss–Hermite nodes and weights, normalized so that $\sum_i w_i g(z_i) \approx \mathbb{E}[g(Z)]$ for $Z \sim \mathcal{N}(0, 1)$. Each map is standardized at the population level, $\tilde{T}(z) = (T(z) - \mu_T)/\sigma_T$, with $\mu_T = \sum_i w_i T(z_i)$ and $\sigma_T^2 = \sum_i w_i (T(z_i) - \mu_T)^2$.

Forward spectrum. The orthonormal Hermite coefficients are

$$c_k = \frac{1}{\sqrt{k!}} \sum_{i=1}^{m} w_i \, \tilde{T}(z_i) \, \mathrm{He}_k(z_i), \qquad E_k = c_k^2,$$

so that $\sum_k E_k = 1$ and $\eta = 1 - E_1$. For cubic and quad-cubic maps the expansion terminates at degree 3, so $E_{\geq 4} = 0$ exactly; for the quintic map, $z^5 = \mathrm{He}_5 + 10\,\mathrm{He}_3 + 15\,\mathrm{He}_1$ gives $E_5/E_3 = 1/5$ and $E_{\geq 4}/\eta = 1/6$ exactly.

Strength calibration. For each family and target $\eta^\star \in \{0.05, 0.10, 0.15, 0.20, 0.30\}$, the strength a is the root of $\eta(a) - \eta^\star$, found with Brent's method on the quadrature-based $\eta(a)$.

Inverse spectrum. The inverse map expresses Z as a function of $X = \tilde{T}(Z)$, whose law μ_X is non-Gaussian, so Hermite polynomials are not orthogonal under it. We construct polynomials through degree K that are orthonormal under the quadrature approximation to μ_X: on the pushed-forward nodes $x_i = \tilde{T}(z_i)$ (which carry weights w_i under μ_X), we form the weighted monomial matrix $A_{ik} = \sqrt{w_i}\, x_i^k$ for $k = 0, \ldots, K$ with $K = 10$ and take its thin QR decomposition $A = QR$. The entries of Q are weighted polynomial evaluations, $Q_{ik} = \sqrt{w_i}\, q_k(x_i)$, where the polynomials $q_0, \ldots, q_K$ are orthonormal under the discrete quadrature measure. The inverse coefficients and energies are

$$d_k = \sum_{i=1}^{m} \sqrt{w_i}\, Q_{ik}\, z_i \approx \mathbb{E}[Z\, q_k(X)], \qquad E_k^{\mathrm{inv}} = d_k^2.$$

Because Z and X are both standardized, $d_1^2 = \mathrm{Corr}(Z, X)^2 = c_1^2$: the linear energy, and hence η, is the same in both directions, while the way the remaining η is spread across higher degrees can differ sharply.

Projection residual. We report the unexplained energy $R_K = 1 - \sum_{k=0}^{K} d_k^2$ rather than assuming the polynomial basis is complete. This matters for the exponential family: X is then a shifted lognormal, whose moment problem is indeterminate, so polynomials need not be dense in $L^2(\mu_X)$ and R_K need not vanish as $K \to \infty$.

Curvature. For the standardized map, curvature is $\kappa(z) = |\tilde{T}''(z)|/(1 + \tilde{T}'(z)^2)^{3/2}$, with derivatives from central finite differences ($h = 10^{-4}$), summarized as $\bar{\kappa} = \sum_i w_i\, \kappa(z_i) = \mathbb{E}_Z[\kappa(Z)]$. Because the graph of $\tilde{T}^{-1}$ is the reflection of the graph of $\tilde{T}$ across the diagonal, and reflection preserves curvature, $\kappa_{\tilde{T}^{-1}}(\tilde{T}(z)) = \kappa_{\tilde{T}}(z)$ pointwise. Since $X = \tilde{T}(Z)$, the expectation is also the same in both directions, so $\bar{\kappa}$ is direction-invariant.

A.3 Full Higher-Order Spectral Energy

$$E_{k \geq 4} = \eta - E_2 - E_3$$

The spectral allocation table reports the fraction of nonlinear energy remaining after removing the quadratic and cubic components. This isolates the higher-degree portion of the spectrum beyond the quadratic and cubic components, allowing transformations with similar total nonlinearity to be distinguished by how much of their nonlinear structure is allocated to more highly ordered terms.

Table 4: Fraction of nonlinear energy remaining after projection through degree 3, $(\eta - E_2 - E_3)/\eta$. For the forward Hermite expansion this corresponds to energy at degrees $k \geq 4$; for the inverse expansion it also includes the finite-projection residual.

Transform	η	Forward $E_{k\geq 4}/\eta$	Inverse $E_{k\geq 4}/\eta$
Cubic	0.05	$\leq 10^{-10}$	0.48792
Cubic	0.1	$\leq 10^{-10}$	0.62968
Cubic	0.15	$\leq 10^{-10}$	0.70795
Cubic	0.2	$\leq 10^{-10}$	0.76097
Cubic	0.3	$\leq 10^{-10}$	0.83275
Quintic	0.05	0.16667	0.79243
Quintic	0.1	0.16667	0.85862
Quintic	0.15	0.16667	0.88683
Quintic	0.2	0.16667	0.90432
Quintic	0.3	0.16667	0.92738
Exp	0.05	0.00085	0.02633
Exp	0.1	0.00348	0.09237
Exp	0.15	0.00800	0.18213
Exp	0.2	0.01457	0.28360
Exp	0.3	0.03451	0.48914
Quad	0.05	$\leq 10^{-10}$	0.05274
Quad	0.1	$\leq 10^{-10}$	0.14603
Quad	0.15	$\leq 10^{-10}$	0.24097
Quad	0.2	$\leq 10^{-10}$	0.32682
Quad	0.3	$\leq 10^{-10}$	0.46057

B Additional Results

B.1 Degradation at Other Nonlinearity Levels

Here we present extended results comparing the test R^2 values of the default neural network model for the forward and inverse maps with more values of η. See Figure 4. Compare with the results in Section 3.1. Generally, higher values of η lead to larger performance penalties, especially at smaller samples.

B.2 Where in the Signal Degradation Occurs

We ran an additional experiment to see where the degradation was occurring. We divided the data into 3 quantile based regions, the lowest 10%, the middle 80%, and the largest 10%. These are all based on the noiseless values, so degradation in the tails is not a byproduct of more noise. We found that while degradation was more concentrated in the tails than in the middle, the middle still had substantial degradation.

As Figure 5 shows, the tails have greater degradation per point, but the center region still accounts for a large impact on the overall degradation. Table 5 reports the proportion of degradation that the central region accounts for.

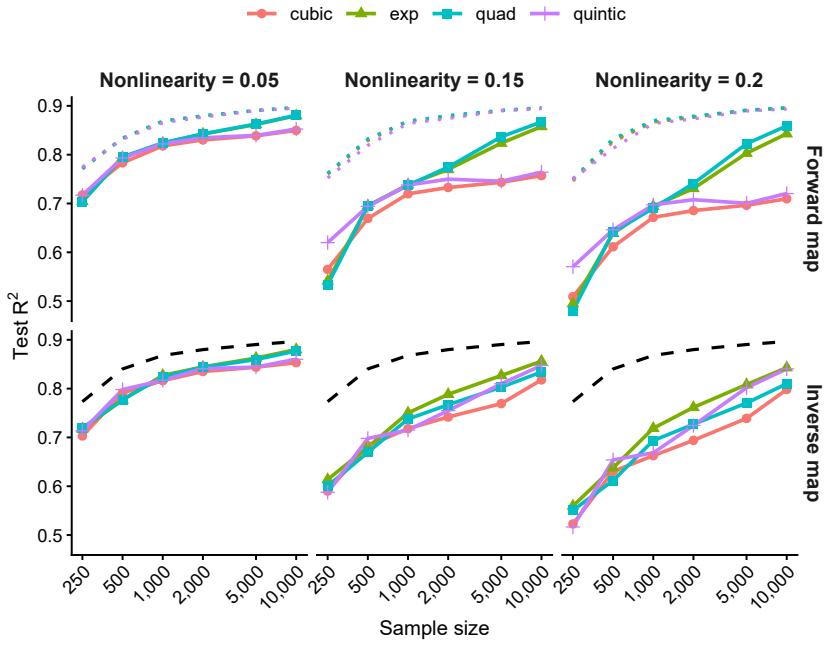

Figure 4: Test R^2 based on mapping, level of nonlinearity, and sample size for different transforms. The dotted and dashed curves represent the control. For the forward map, the control represents R^2_{TT} and is the control for each individual transform. The control for the inverse is R^2_{BB} and is thus identical for every transform.

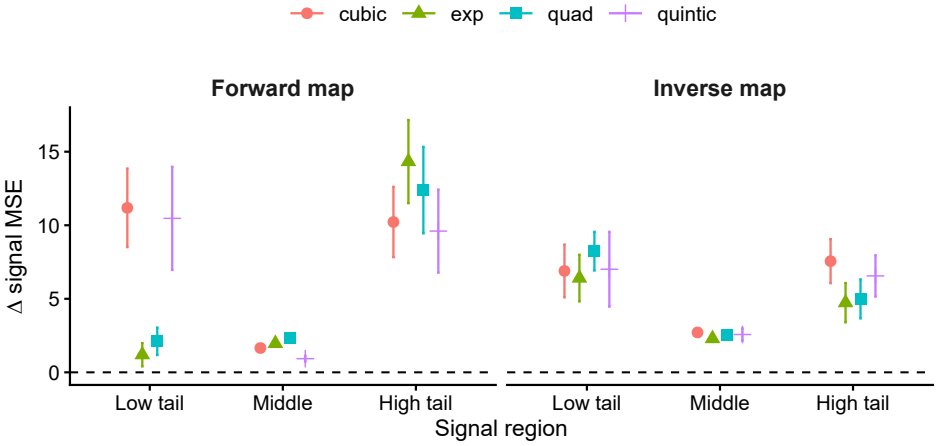

Figure 5: Increase in prediction MSE induced by representation mismatch at $\eta = 0.30$, stratified by quantiles of the noiseless signal. Points show paired mean differences relative to the corresponding matched condition and error bars show 95% confidence intervals. Although the per-observation penalty is largest in the tails, positive excess error remains throughout the central 80% of the signal distribution in both map directions. The experiment followed the main pipeline at $n = 500$.

Table 5: Share of mismatch-induced SSE arising from the central signal region. Values are means across paired repetitions ± SE. The complementary share arises from the two tails combined. The experiment followed the main pipeline at $n = 500$.

Transform	Forward Map	Inverse Map
Cubic	0.3659 ± 0.0295	0.6188 ± 0.0254
Exp	0.5642 ± 0.0415	0.6347 ± 0.0243
Quad	0.6042 ± 0.0378	0.6210 ± 0.0202
Quintic	0.2674 ± 0.02999	0.6281 ± 0.0359

B.3 Transforming Active and Inactive Predictors

We examined whether the representation penalty depends on which predictors are transformed as well as the quantity of transformed predictors. This experiment was run on $n = 500$ and only looks at the largest nonlinear strengths for each transform, $\eta = 0.3$. There is an approximately linear relationship between R^2 and transformed active predictors. Across all transforms and both map directions, test R^2 varied almost perfectly linearly with the number of transformed predictors, with all within-condition linear fits achieving $R^2 \geq .995$. This behavior is consistent with the data generating process as it is just a linear sum of the active predictors plus noise.

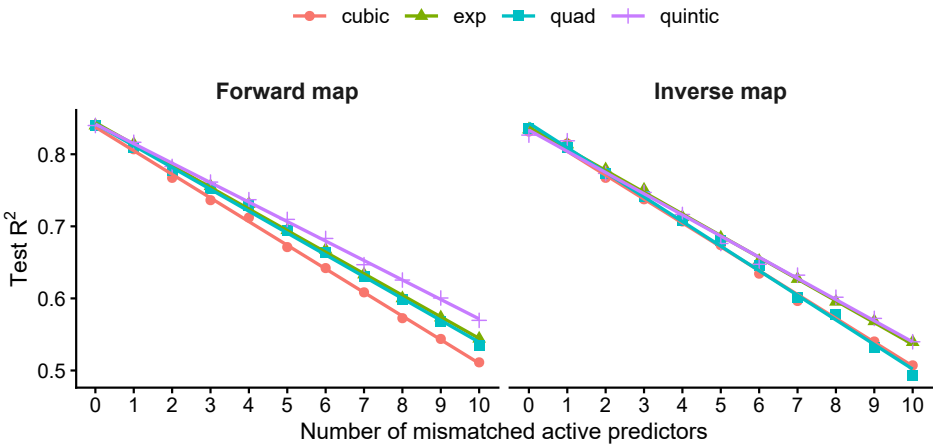

Figure 6: Test R^2 as a function of number of mismatched active predictors for each transform family using main experimental design with $n = 500$. In the forward condition (BT), a predictor is mismatched when the predictor is untransformed; For the inverse map (TB), a predictor is mismatched when a transform is applied to the predictor.

Transforming inactive predictors had a negligible additional effect on predictive performance. Restricting the comparison to conditions with ten transformed active predictors, for which matched inactive-transformed and inactive-untransformed conditions were available, the mean paired difference was

$$R^2_{\text{inactive transformed}} - R^2_{\text{inactive untransformed}} = 0.00201 \pm 0.00174,$$

where the reported uncertainty is the standard error across the paired runs.

B.4 CIs for Figure 1

Figure 7 reproduces Figure 1 performance comparisons with 95% CIs. These intervals summarize between-repetition variability in test R^2 and therefore provide a direct view of the stability of the reported mean performance differences. Variability is greatest at smaller sample sizes and generally contracts as the sample size increases. The confidence intervals also show that the broad learning-

curve patterns reported in the main text persist after accounting for simulation variability, although some individual conditions exhibit substantial overlap.

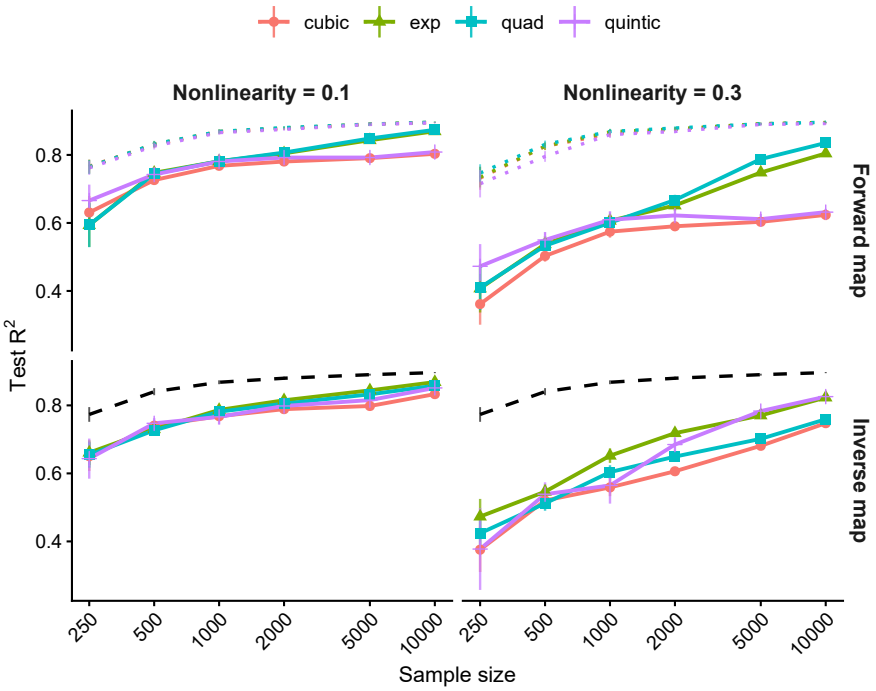

Figure 7: Test R^2 across sample size for transformations matched at $\eta = 0.10$ and $\eta = 0.30$. Solid curves show mismatched conditions: BT for the forward map and TB for the inverse map. In the forward panels, transform-specific dotted curves show the corresponding TT matched controls; in the inverse panels, the black dashed curve shows the common BB control. All groups have associated 95% confidence intervals. Although transformations within each panel have the same population nonlinearity η, their finite-sample learning curves differ across transformation families and map directions.

C Is the Penalty a Pipeline Artifact?

C.1 Population Standardization

To ensure that the degradation was not simply a byproduct of the standardization method, an exact population standardization control was instituted to verify that the effect persisted.

The effect clearly persists when the population level standardization is implemented indicating that the degradation is not a byproduct of the method of standardization.

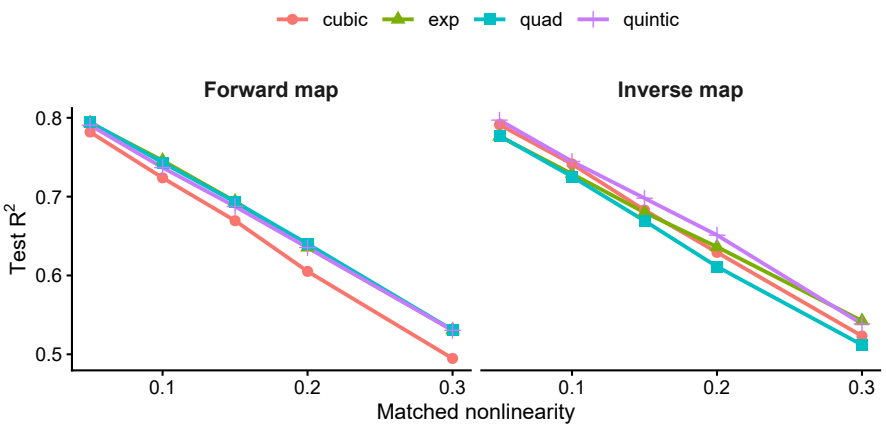

Figure 8: Test R^2 at $n = 500$ when base and transformed representations are standardized using population rather than sample-specific moments. The representation penalties persist under this alternative construction, indicating that the primary effect is not explained by the empirical standardization procedure.

C.2 Optimization and Capacity Sweeps

We disable early stopping and vary the training budget from 250 to 2000 epochs at $n = 500$. Test R^2 changes only slightly across budgets, and the penalties persist in both directions (Figure 9). Increasing the network parameter count by a factor of 8.6 reduces the penalties by only 6–14% (Table 6). Every fit reaches training $R^2 \geq 0.99$ in all four conditions, so the networks nearly interpolate the training data. The representation penalties persist despite longer training, increased capacity, and near-perfect training fit.

Table 6: Representation penalties under increasing MLP width at $n = 500$. Each network is trained for 2000 epochs without early stopping, with L2 penalty 10^{-4}, learning rate 10^{-3}, and batch size 64 fixed. Entries are mean paired $\Delta R^2 \pm$ SE over 25 repetitions. Every fit reaches training $R^2 \geq 0.99$ in all four conditions, indicating that substantial test-performance penalties remain despite near-perfect training fit. An $8.6\times$ increase in parameters reduces the penalties by only 6–14%, and all penalties remain large.

			$\eta = 0.20$		$\eta = 0.30$	
Family	Layers	Params	Δ_{forward}	Δ_{inverse}	Δ_{forward}	Δ_{inverse}
Cubic	64–32	5.4k	0.231 ± 0.013	0.239 ± 0.013	0.353 ± 0.016	0.379 ± 0.020
	128–64	14.8k	0.221 ± 0.010	0.227 ± 0.011	0.332 ± 0.013	0.359 ± 0.016
	256–128	46.1k	0.206 ± 0.010	0.220 ± 0.012	0.309 ± 0.012	0.329 ± 0.015
Quintic	64–32	5.4k	0.179 ± 0.015	0.190 ± 0.018	0.270 ± 0.018	0.301 ± 0.032
	128–64	14.8k	0.176 ± 0.013	0.173 ± 0.017	0.258 ± 0.018	0.269 ± 0.027
	256–128	46.1k	0.157 ± 0.013	0.173 ± 0.016	0.233 ± 0.017	0.262 ± 0.025
Exp	64–32	5.4k	0.178 ± 0.012	0.213 ± 0.014	0.272 ± 0.017	0.301 ± 0.016
	128–64	14.8k	0.175 ± 0.008	0.186 ± 0.011	0.260 ± 0.011	0.263 ± 0.014
	256–128	46.1k	0.167 ± 0.009	0.190 ± 0.012	0.249 ± 0.014	0.266 ± 0.016
Quad–cubic	64–32	5.4k	0.180 ± 0.011	0.242 ± 0.014	0.279 ± 0.015	0.363 ± 0.018
	128–64	14.8k	0.176 ± 0.008	0.218 ± 0.013	0.269 ± 0.010	0.328 ± 0.016
	256–128	46.1k	0.169 ± 0.008	0.222 ± 0.012	0.256 ± 0.011	0.324 ± 0.017

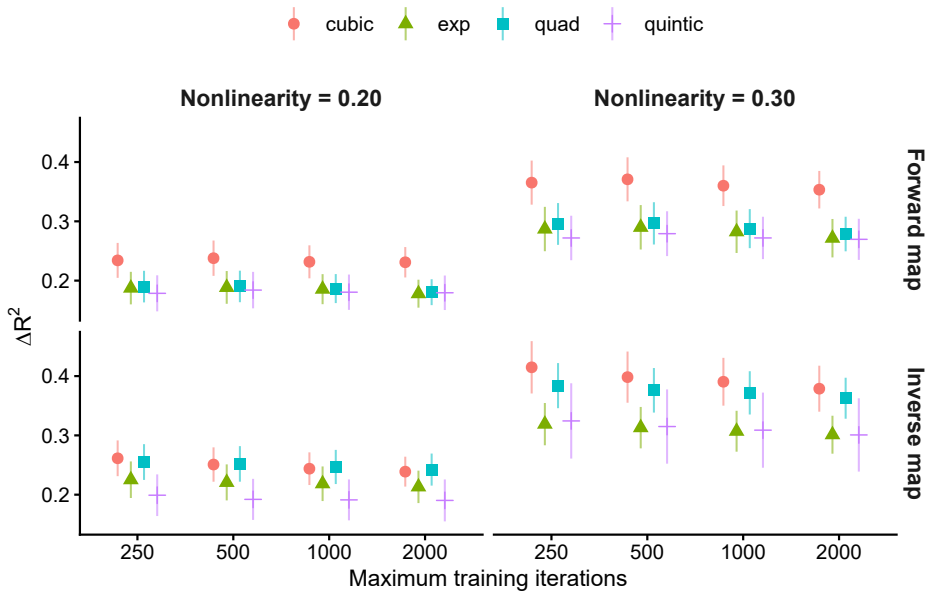

Figure 9: Representation degradation as a function of optimization budget for a fixed 64-32 MLP at $n = 500$. Forward degradation is $\Delta R^2 = R^2_{TT} - R^2_{BT}$, and inverse degradation is $\Delta R^2 = R^2_{BB} - R^2_{TB}$. Points show mean paired degradation across 25 repetitions, with error bars denoting 95% confidence intervals computed as mean ± 1.96 SE. The number of training epochs is varied from 250 to 2000 while the architecture and other optimization hyperparameters are held fixed. Representation penalties persist across optimization budgets, indicating that the observed mismatch effect is not eliminated by substantially increasing the training-iteration budget.

C.3 XGBoost Invariance

XGBoost serves as a check on the data pipeline. Each tree splits one feature at a threshold, and a strictly monotone transformation of a feature changes the threshold values but not the partitions of the data that splits can produce. We therefore expect approximate invariance under this implementation; a sizable penalty would warrant checking both the paired data construction and the learner implementation. Figure 10 confirms the expected behavior: across all families, nonlinearity levels and sample sizes, $|\Delta R^2|$ stays below about 4×10^{-4} in both directions.

D Other Learners

To test whether the representation penalty is specific to the scikit-learn MLP, we repeat the main comparison with two other learners (configurations in Table 3).

D.1 ResNet Degradation

The residual MLP (Figure 11, $n = 500$) shows the same qualitative sensitivity as the primary MLP: penalties grow with matched nonlinearity, and they differ across transformation families and between the forward and inverse directions. Because the ResNet uses a single fixed configuration without hyperparameter search, this also shows that the penalty does not depend on the tuning procedure.

D.2 TabICLv2 Degradation

The pretrained in-context model (Figure 12, $n = 2000$) is much more robust. Its test R^2 varies over a far narrower range than the MLP's, and the inverse penalty in particular is small. It is not invariant, however: small differences across transformation families and directions remain. TabICLv2's

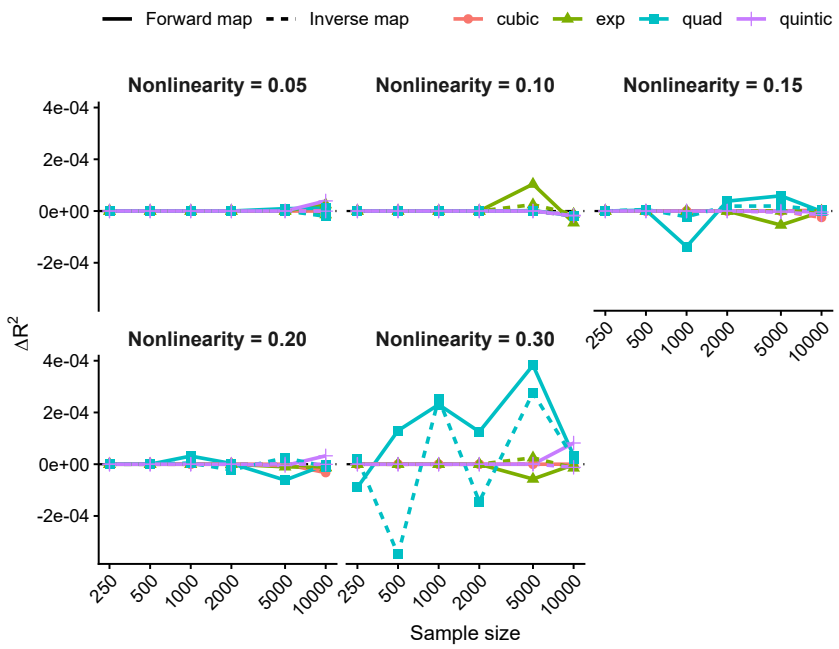

Figure 10: XGBoost representation penalties across sample size, transformation family, and matched nonlinearity. Forward degradation is $\Delta R^2 = R^2_{TT} - R^2_{BT}$; inverse degradation is $\Delta R^2 = R^2_{BB} - R^2_{TB}$. Values remain essentially zero across conditions, consistent with the near-invariance of axis-aligned tree partitions to strictly monotone feature transformations. The small remaining differences are consistent with numerical and implementation effects, although we do not isolate their source.

default preprocessing applies a Yeo–Johnson power transform to half of its eight ensemble members, partially Gaussianizing the observed features. Consistent with Section 3.4, this helps most in the inverse condition, where the power transform moves U toward the base coordinates, and cannot help in the forward condition, where the inputs are already Gaussian.

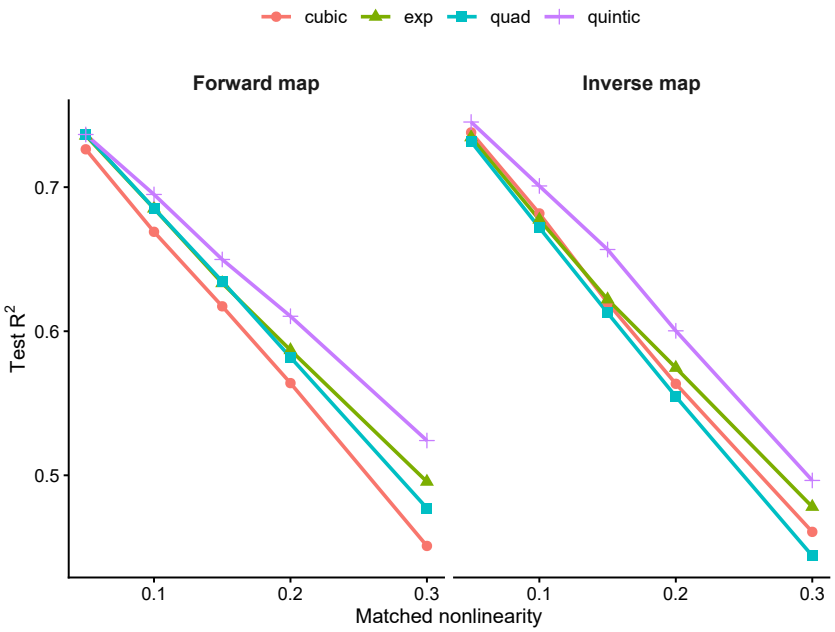

Figure 11: Test R^2 for forward and inverse mismatch conditions using the ResNet tabular model at $n = 500$. Performance declines as matched population nonlinearity increases, reproducing the qualitative representation sensitivity observed for the primary MLP while preserving transformation-family and direction-specific differences.

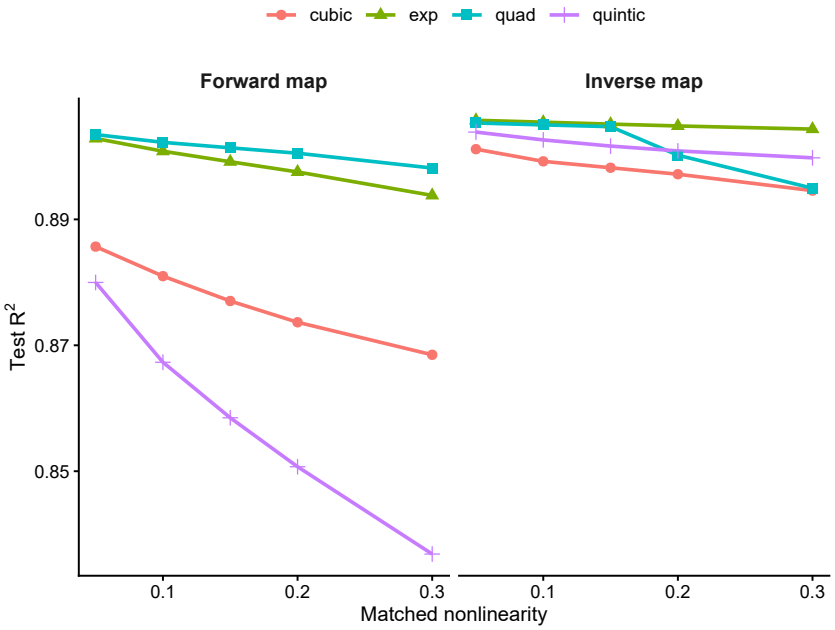

Figure 12: Test R^2 for forward and inverse mismatch conditions using TabICLv2 at $n = 2000$. TabICLv2 is substantially more robust than the primary MLP, particularly in the inverse condition, while remaining sensitive to representation in a transformation- and direction-dependent manner.