Feature Importance for Model Fit: Decomposing Mean Squared Error in Nonlinear Regression

January 20, 2026

Abstract

We study the problem of attributing explained predictive fit in a nonlinear regression model to individual input variables. We derive an exact, additive decomposition of explained fit by applying the fundamental theorem of calculus along a path in input space, expressing the reduction in expected loss relative to a baseline prediction as a sum of feature-level contributions defined by integrated loss gradients.

The resulting attribution is global, additive, and model-conditional. It generalizes Euler-based decompositions of explained signal strength beyond linear regression and provides a principled framework for understanding how predictive performance is generated by inputs in nonlinear models under explicit and interpretable assumptions.

We also derive standard errors for the attributed contributions, enabling statistical assessment of variation in feature importance across samples or over time. The computations for both the contributions and their standard errors have predictable and moderate cost, remaining feasible even for high-dimensional models.

www.ludgerhentschel.com ludger@ludgerhentschel.com

Contents

Acknowledgements
For helpful comments, I am grateful to Nishant Gurnani and Shubham Jaiswal.

1 Introduction

In complex predictive systems, two central questions arise naturally: whether a model performs well under a given accuracy measure, and how input features contribute to that performance. Monitoring predictive accuracy is well understood. This paper addresses the second question by developing a principled method for attributing predictive fit to input features.

We study feature importance in nonlinear regression models of the form

$$\widehat{y}(x) = \widehat{\alpha} + f(\widehat{\theta}, x), \tag{1}$$

where $\widehat{\alpha}$ is an intercept, x denotes a vector of input features, and $\widehat{\theta}$ denotes fitted parameters. We do not assume access to the parameters themselves, but treat the trained model as defining a fixed prediction function $f(\widehat{\theta}, \cdot)$ that can be evaluated at admissible inputs. We focus on settings in which each observation produces a scalar prediction.[1]

Our objective is to attribute explained predictive fit back to individual input variables for a fixed fitted model. We derive an additive, model-conditional decomposition of predictive fit for nonlinear prediction functions that parallels the Euler decomposition of explained signal strength in linear regression. We can compute the resulting attribution in predictable and manageable time, even for high-dimensional models, enabling regular and repeated analysis with accompanying measures of statistical uncertainty.

At a high level, the approach proceeds as follows. We measure predictive fit as the reduction in expected loss relative to a fixed baseline prediction. To attribute this reduction to input variables in a nonlinear model, we trace how the loss evolves as inputs move from the baseline to their realized values along a simple path in input space. Integrating the loss gradient along this path yields an exact additive decomposition of explained fit, generalizing Euler-style decompositions to nonlinear settings.[2]

In addition to the contributions to model fit, we also derive the corresponding standard errors that reflect sampling variability in the data. These standard errors are computed from observation-level contributions and quantify uncertainty in the attribution for a fixed fitted model. They permit us to assess, in statistical terms, whether observed variation in feature contribu-

[1] We can accommodate multivariate outputs by defining a scalar loss on the output vector, but we do not pursue this extension here.

[2] This construction is analogous to a work integral in physics. When a force field is the gradient of a scalar potential, the total change between two points equals the endpoint difference, as formalized by the fundamental theorem of line integrals. Here, the loss gradient plays the role of the force field. While the total change in model fit is fixed, a path is required to determine how that change is allocated across input dimensions, which is precisely the attribution problem we study.

tions across samples or over time is plausibly attributable to noise or instead reflects changes in predictive relevance.

The framework applies to prediction functions that are differentiable almost everywhere along the path from baseline to realization. This includes piecewise-smooth and piecewise-constant functions and encompasses a broad class of models, including linear and nonlinear regressions (with or without regularization), generalized additive models, tree-based methods and their ensembles, neural networks, kernel methods, and state-space or regime-switching models, provided predictive fit is evaluated using a differentiable loss such as quadratic loss.[3]

The framework does not apply to models with intrinsically discrete or combinatorial inputs, to models whose outputs are stochastic rather than deterministic functions of the inputs, or to models whose internal representations do not admit meaningful infinitesimal input variation. Examples include rule-based systems, discrete optimization models, and generative models defined through sampling.

Related work on feature importance largely addresses two distinct questions. In statistics, there is substantial work on assessing how important a feature is for explaining the data y, without conditioning on a specific fitted model $f(\widehat{\theta}, \cdot)$; see Budescu (1993) for a review. Such measures are useful for exploratory analysis and model development, but they generally say little about how much a feature contributes to the fit of a particular fixed model.

In regression analysis, partial R^2 is a classical measure of feature importance defined as the reduction in model fit when a regressor is removed and the remaining coefficients are re-estimated. While partial R^2 focuses on model fit rather than individual predictions, it compares fit across alternative models rather than decomposing the realized fit of a given fitted model. The framework developed here instead addresses this latter decomposition problem.

In machine learning, a separate literature focuses on how input variables contribute to individual predictions $\widehat{y}$, sometimes aggregating these contributions across observations to obtain global summaries; see Guidotti, Monreale, Ruggieri, Turini, Giannotti, and Pedreschi (2018) and Molnar (2022) for reviews. These analyses are valuable for understanding how models use their inputs and for explaining individual predictions. However, sensitivity of predictions to features generally says little about which features contribute to overall model fit or predictive performance. Features can make large

[3] Although tree-based models are not differentiable everywhere, the path integral remains well defined because points of non-differentiability occur on sets of measure zero.

contributions to predictions while simultaneously degrading predictive fit, a concern that is particularly acute outside the training sample, once a model has moved from development to production. Contributions to predictions and contributions to model fit therefore measure fundamentally different quantities. This paper focuses on the latter.

The remainder of the paper proceeds as follows. Section 2 formalizes our notion of feature importance and derives the corresponding attribution. Section 3 provides a detailed comparison to other prominent feature-importance measures and section 4 concludes. Appendix A derives standard errors for the contributions and appendix B presents the computational details underlying the attribution.

2 Attributing Explained Fit

Our objective is to decompose a scalar measure of predictive fit into additive, feature-level contributions. The resulting attribution answers the question: *How much does each input feature contribute to the model's overall predictive performance?* This perspective parallels Euler decompositions of explained signal strength in linear regression and marginal risk contributions in portfolio attribution.

2.1 Our Notion of Feature Importance

In this paper, feature importance refers to the contribution of an input variable x_j to the overall predictive performance within the fitted model actually used. We call this the feature's contribution to model fit, or simply its feature contribution. We measure feature contributions relative to a specific baseline model and a specific scalar measure of fit, such as explained variance or reduction in expected loss relative to a baseline predictor.

This notion of importance is global, additive, and model-conditional. It attributes realized predictive performance within the fitted model actually used, rather than hypothetical performance under feature removal, refitting, or intervention. A feature is important if it materially contributes to the model's predictive success, even if other features could potentially substitute for it under alternative specifications or refitted models.

We do not interpret feature importance as local sensitivity of predictions to marginal input changes, as robustness of performance under feature perturbation or removal, as explanation of individual predictions, or as causal influence on the data-generating process. These notions address different questions and we compare them to the present framework in section 3.

By fixing attention on contribution to realized predictive performance within a fitted model, the proposed framework exploits structure that more

general attribution methods deliberately ignore. This restriction enables exact additivity, clear interpretation, and computational efficiency, even in highly nonlinear regression models with many features.

2.2 Setup

Let $y \in \mathbb{R}^N$ denote a vector of observed outcomes with finite variance, and let $X \in \mathbb{R}^{N \times K}$ denote a matrix of input features. Each row $x_i \in \mathbb{R}^K$ represents the feature vector for observation i. Importantly, X and y may differ from the training sample used to estimate the model parameters $\widehat{\theta}$.

Purely to facilitate interpretation, we center and standardize all input features in sample so that $\mathbb{E}[x_{ij}] = 0$ and $\mathrm{Var}(x_{ij}) = 1$ for each feature j. These normalizations are common for regularized models and apply to continuous inputs as well as binary indicators and group dummies.[4]

Let $\widehat{y}(x) = \widehat{\alpha} + f(\widehat{\theta}, x)$ denote the fitted prediction produced by a regression or prediction model for a single observation with input x. For brevity, we suppress the fitted parameters $\widehat{\theta}$, which we treat as fixed throughout. Applied row-wise to X, the fitted model produces the N-vector of predictions

$$\widehat{y}(X) = \widehat{\alpha}\,\mathbf{1} + f(X), \tag{2}$$

where $f(X) = \big(f(x_1), \ldots, f(x_N)\big)$ collects the nonlinear component.

We evaluate model fit using a differentiable loss function. We write $\ell(y_i, \widehat{y}_i)$ for the scalar, per-observation, local loss and apply it element-wise across observations. For vectors $y, \widehat{y} \in \mathbb{R}^N$, we interpret

$$\ell(y, \widehat{y}) = \big(\ell(y_1, \widehat{y}_1), \ldots, \ell(y_N, \widehat{y}_N)\big). \tag{3}$$

We write $\mathcal{L}(\widehat{y}) = \mathbb{E}[\ell(y, \widehat{y})]$ for the corresponding sample-average, global loss.

We center outcomes so that $\mathbb{E}[y_i] = 0$. The constant predictor $\overline{y} = \widehat{\alpha}$ therefore serves as a natural baseline prediction, independent of the choice of baseline input x_0.[5] The baseline remains fixed and is not refit as part of the attribution exercise.

We define explained fit as the reduction in aggregate loss relative to this baseline,

$$\Delta\mathcal{L} = \mathcal{L}(\overline{y}) - \mathcal{L}(\widehat{y}(X)) = \mathbb{E}\big[\ell(y, \overline{y})\big] - \mathbb{E}\big[\ell(y, \widehat{y}(X))\big]\,. \tag{4}$$

[4] For discrete inputs, intermediate values encountered along the attribution path are analytical constructs; attribution reflects the contribution of moving between regimes rather than sensitivity to infinitesimal variation.

[5] Throughout, we take the baseline prediction to be a fixed, low-dimensional reference model, such as a constant mean or a small set of group-specific means defined by exogenous indicators. More flexible baselines are mechanically admissible but blur the interpretation of explained fit.

For squared error loss, $\Delta\mathcal{L}$ is proportional to R^2. More generally, the formulation applies to any differentiable loss.

2.3 A Path-Integral Decomposition of Explained Fit

In order to compute the change in loss $\Delta\mathcal{L}$, we apply the fundamental theorem of calculus along a path in input space that connects a baseline input to the realized input. We do this for each observation and then average across observations.

Let x_0 denote a baseline input vector, which we take to be $x_0 = 0$ without loss of generality under centering in the presence of the intercept α.[6] The baseline input serves as a common starting point for the path in input space for all observations. It also represents an uninformed model analogous to an intercept-only linear regression.

For each input x, we adopt a straight-line path

$$x(t) = x_0 + t(x - x_0), \quad t \in [0, 1]. \tag{5}$$

This is the same path used by the integrated gradients (IG) methodology of Sundararajan, Taly, and Yan (2017).[7] As Sundararajan, Taly, and Yan (2017) emphasize, this path choice has several desirable properties: it treats all input features symmetrically, introduces no additional modeling assumptions, and corresponds to a uniform interpolation from the baseline input to the realized input.

Alternative paths may be appropriate in settings with constrained input domains, ordered information structures, or semantically grouped features. In all cases, the path-integral construction remains valid; only the convention for allocating interaction effects changes. In linear models, the choice of path is immaterial, because homogeneity collapses the path integral to the Euler decomposition. In nonlinear models, the straight-line path provides a simple and transparent generalization that preserves additivity while minimizing arbitrariness.

For notational simplicity, we write $\ell(y, f(x))$ rather than $\ell(y, \widehat{\alpha} + f(x))$. The intercept $\widehat{\alpha}$ is constant in x and cancels from all loss gradients and path integrals.

[6] In some applications, domain knowledge may suggest a baseline input $x_0 \neq 0$ that represents a state of minimal or neutral information. With an explicit intercept α, such cases should be rare.

[7] Integrated gradients decompose individual predictions, whereas we decompose model fit. The two approaches share the same path in input space but apply it to fundamentally different target functionals.

For a fixed observation, the fundamental theorem of calculus says that the change in local loss relative to the baseline satisfies

$$\ell(y, f(x)) - \ell(y, f(x_0)) = \int_0^1 \frac{d}{dt} \ell(y, f(x(t)))\, dt \tag{6}$$

$$= \int_0^1 (x - x_0)^\top \nabla_x \ell(y, f(x(t)))\, dt. \tag{7}$$

This identity is exact. It expresses the change in loss as an integral of the loss gradient accumulated along the path from the baseline input to the realized input.[8] Equation (7) decomposes the change in loss from the baseline to the realized prediction. Explained fit corresponds to the negative of this quantity, which is why the input-level contributions defined below carry a leading minus sign.

Rewriting the inner product yields an additive decomposition across input coordinates,

$$\ell(y, f(x)) - \ell(y, f(x_0)) = \sum_{j=1}^{K} (x_j - x_{0j}) \int_0^1 \frac{\partial}{\partial x_j} \ell(y, f(x(t)))\, dt. \tag{8}$$

This identity holds point-wise for each observation y_i. Averaging across observations produces a global, additive decomposition of explained fit,

$$\Delta\mathcal{L} = \sum_{j=1}^{K} C_j, \tag{9}$$

where we define the contribution of input x_j as

$$C_j = -\mathbb{E}\left[(x_j - x_{0j}) \int_0^1 \frac{\partial}{\partial x_j} \ell(y, f(x(t)))\, dt \right]. \tag{10}$$

The path-integral identity in equation (7) decomposes the change in loss from the baseline input to the realized input. Because equation (4) defines explained fit as the negative of this change, a reduction in loss, we introduce a leading minus sign so that positive contributions correspond to features that improve predictive performance.

By construction, the contributions C_j sum exactly to the total reduction in expected loss. The decomposition is model-conditional and attributes

[8] The identity in equation (7) is also an instance of the fundamental theorem of line integrals. Because the vector field $\nabla_x \ell(y, f(x))$ is the gradient of a scalar function, the *total* change in loss between the baseline and the realized input is path independent and equals the endpoint difference $\ell(y, f(x)) - \ell(y, f(x_0))$ for any smooth path connecting x_0 to x. Path dependence arises only when we seek to allocate this total change across input coordinates, as we do here.

explained fit to the inputs of the fitted model actually used.

The quantities C_j are contributions to predictive fit and can be interpreted as measures of feature importance for model accuracy. They are not shares of a bounded resource. In additive decompositions of positive quantities, individual components need not lie in the $[0, 1]$ interval and may be negative.[9] A negative contribution C_j indicates that, conditional on the fitted model, feature j reduces predictive fit on average.

A negative contribution C_j does not imply that predictive performance would improve if the corresponding feature were removed. The decomposition is model-conditional: it attributes realized fit within a fixed fitted model. Removing a feature and refitting produces a different model with different fitted scores and a different attribution. Moreover, contributions are jointly determined and need not correspond to marginal gains from inclusion. Dropping one component generally changes the contributions of all others. Euler-style contributions should be interpreted as an analysis of how predictive fit is generated within the fitted model, not as a prescription for feature selection.

The path-integral identity requires that the composite mapping $x \mapsto \ell(y, f(x))$ be differentiable almost everywhere along the chosen path in input space. Since common loss functions used to evaluate predictive fit are differentiable almost everywhere, the main practical requirement is that the prediction function $f(x)$ be differentiable almost everywhere along the path. This is not a severe restriction in practice: isolated points of non-differentiability do not affect the integral and therefore do not invalidate the decomposition. In particular, the framework accommodates piecewise-constant or piecewise-smooth prediction functions, such as those arising from tree-based models, spline-based models, or rectified linear unit (ReLU) networks.

In principle, any smooth path connecting x_0 to x yields a valid decomposition of the change in loss. The need to specify a path is unavoidable in the absence of linearity or homogeneity and reflects the fact that nonlinear models do not possess a canonical additive representation of the fitted signal. Path dependence is not a defect of the attribution, but a consequence of nonlinearity.

When the fitted model is approximately additive, or when interaction effects are weak, different smooth paths yield similar attributions. When interactions are strong, however, path dependence becomes economically

[9] For $\mathrm{Var}(a + b) = \mathrm{Var}(a) + \mathrm{Var}(b) + 2\,\mathrm{Cov}(a, b)$ we are not concerned that $2\,\mathrm{Cov}(a, b)$ can be a negative component of the total variance.

meaningful: it reflects explicit choices about how interaction effects in the loss are allocated across inputs.

The path adopted here traverses the interior of the hypercube connecting x_0 to x, treating all inputs symmetrically and allocating interaction effects continuously along the interpolation. By contrast, several alternative feature importance measures rely on paths that move along the edges of the hypercube, in which inputs are activated sequentially and interaction effects are implicitly assigned to specific features by construction.

2.4 Squared Error Loss

The standard measure of regression fit is squared error loss,

$$\ell(y_i, \widehat{y}_i) = (y_i - \widehat{y}_i)^2. \tag{11}$$

Under this loss, the derivative with respect to the fitted value is linear in the residual,

$$\frac{\partial \ell(y_i, \widehat{y}_i)}{\partial \widehat{y}_i} = -2(y_i - \widehat{y}_i). \tag{12}$$

Substituting this expression into the general definition of feature contributions yields

$$C_j = 2\,\mathbb{E}\left[(x_j - x_{0j}) \int_0^1 \big(y - f(x(t))\big) \frac{\partial f(x(t))}{\partial x_j}\, dt\right]. \tag{13}$$

The factor $(x_j - x_{0j})$ captures variation in the input, $\partial f(x(t))/\partial x_j$ is a local signal sensitivity, and $(y - f(x(t)))$ is the residual along the path.

Under squared error loss, the contribution C_j measures how much variation in input x_j, relative to the baseline x_{0j}, aligns with reductions in squared prediction error along the path from the baseline input to the realized input. Inputs generate larger positive contributions when their variation coincides with directions in input space along which the fitted signal both changes strongly and reduces residual error.

Appendix B summarizes the computation of this attribution. The algorithm computes a global decomposition of explained fit by averaging these path-integral contributions across observations. In practice, the integral is evaluated numerically using a finite quadrature scheme; additivity holds up to numerical integration error.

It is important to note that the numerical integration evaluates a one-dimensional path integral. We do not numerically approximate the high-dimensional regression function. The distinction affects speed and accuracy,

especially in high dimensions K. Moreover, we average the path across a potentially large number of observations N, This further reduces small numerical errors that may affect individual path integrals. Straightforward numerical integration approaches, like quadrature on a fixed grid, are likely to be very effective in this application.

2.5 Connection to Euler Decomposition

We can interpret this construction as an Euler-style decomposition, a concept that may be familiar from portfolio risk attribution. To clarify the connection with Euler's theorem, recall that if $g(z)$ is a positively homogeneous function of degree k, then Euler's theorem implies[10]

$$g(z) = \frac{1}{k} \sum_j z_j \frac{\partial g(z)}{\partial z_j} = \frac{1}{k} z^\top \nabla_z g(z). \tag{14}$$

Because this is an exact additive identity, it is natural to interpret $z_j \, \partial g / \partial z_j \, / \, k$ as the contribution of component z_j to $g(z)$. Although Euler's theorem may resemble a gradient-based expansion local to z, it is in fact a global and exact decomposition, valid at all inputs z.

By direct comparison, equation (7) reduces to an Euler decomposition whenever the path integral can be evaluated in closed form and expressed as an endpoint identity. A sufficient condition for this collapse is that the integrand be at most affine in the path parameter, so that integrating along the path introduces no genuinely path-dependent terms.

A prominent case in which this occurs arises under squared error loss for linear regression models. When

$$f(x) = x^\top \beta, \tag{15}$$

the gradient of the loss with respect to inputs satisfies

$$\frac{\partial}{\partial x_j} \ell(y, f(x)) = -2(y - x^\top \beta)\, \beta_j. \tag{16}$$

Along the straight-line path $x(t) = x_0 + t(x - x_0)$, the fitted value $f(x(t))$ varies linearly in t, and the residual $y - f(x(t))$ therefore varies linearly as well. Consequently, the integrand

$$(x_j - x_{0j}) \, \frac{\partial}{\partial x_j} \ell\big(y, f(x(t))\big) \tag{17}$$

[10] See Silberberg (1978) or Tasche (2008), for example.

is affine in t, and integrating it over $t \in [0, 1]$ yields an exact endpoint identity.

Under these conditions, the line integral collapses to an endpoint identity, yielding an Euler decomposition of explained signal strength evaluated at the realized input,

$$\ell(y, f(x)) - \ell(y, f(x_0)) = \sum_{j=1}^{K} (x_j - x_{0j}) \frac{\partial}{\partial x_j} \ell(y, f(x)) \tag{18}$$

$$= -2 \sum_{j=1}^{K} \beta_j (x_j - x_{0j}) (y - x^\top \beta). \tag{19}$$

Linear regression with squared error loss is therefore a special case in which the general path-integral construction reduces exactly to Euler's theorem applied to the explained signal, which is homogeneous of degree one in the inputs. Hentschel (2026) analyzes this case in detail.

Outside this special case, explained fit is no longer a homogeneous function of the inputs, and the endpoint identity implied by Euler's theorem does not apply. The line-integral construction therefore provides an Euler-style decomposition absent homogeneity, recovering changes in fit by integrating marginal contributions along a path from the baseline input to the realized input. Like a standard Euler decomposition, our attribution is additive and unique conditional on the model $f(\cdot)$, the loss $\ell(\cdot)$, the actual inputs x, the baseline inputs x_0, and the linear path connecting x_0 to x.

In many applications, we can express a nonlinear model $f(x)$ as a polynomial or series expansion in x, making the model linear in a set of constructed regressors such as monomials or interaction terms. This representation permits a straightforward Euler decomposition with respect to those constructed components.

However, individual input features generally appear in multiple such terms, for example through x_j^2, $x_j x_k$, or higher-order interactions. A linear Euler decomposition at the level of constructed regressors treats each monomial as a separate component and does not, by itself, yield a coherent attribution at the level of the original input features x_j.

The path-integral attribution developed here operates directly in the original input space. It aggregates all marginal effects of a feature x_j, including those arising through nonlinear and interaction terms, into a single, well-defined contribution to explained fit.

This distinction is not specific to polynomial models. The same issue arises in neural networks and other nonlinear architectures, which are linear in large collections of internal features or activations. Euler-style decompositions

at that level attribute fit to internal components rather than to the original inputs. By operating directly in input space, the path-integral attribution yields feature-level contributions that are invariant to the model's internal representation.

2.6 Weighted Loss Functions

In some applications, we evaluate predictive fit under a weighted loss function. Weighting may reflect heteroskedasticity, differing importance across observations, or the desire to emphasize particular regions of the outcome space. In all cases, weighting affects how we measure prediction errors, but not how features enter the attribution. We always define contributions with respect to the original input variables, with weights entering only through the definition of the loss.

For a generalized least squares formulation, let W denote a weighting matrix applied to residuals. This is commonly a whitening transformation satisfying $W^\top W = \Omega^{-1}$ for some positive definite covariance matrix Ω. In this case, we evaluate predictive fit using the transformed loss

$$\ell_W(y, \widehat{y}) = \|W(y - \widehat{y})\|^2. \tag{20}$$

This formulation is equivalent to evaluating standard squared error loss on pre-whitened outcomes and predictions.

All derivations above apply without modification when ℓ is replaced by ℓ_W. In particular, the path-integral decomposition remains exact, with feature-level contributions defined by integrating weighted loss gradients along the chosen path in input space. Weighting alters the geometry of the output space in which prediction errors are measured, but does not affect the input space along which attribution is performed.

Scalar observation weights arise as a special case when Ω is diagonal. In this case, we can also accommodate more general loss functions. Writing $w_i = \Omega_{ii}^{-1}$, the aggregate weighted loss can be written as

$$\mathcal{L}(f) = \mathbb{E}\left[w_i \, \ell(y_i, \widehat{y}_i)\right]. \tag{21}$$

The attribution framework applies identically in this case.

It is important to distinguish the use of weights in defining predictive fit from the estimation procedure that produced the fitted model. The attribution framework treats the prediction function $f(\widehat{\theta}, \cdot)$ as fixed and does not differentiate through the estimation step. Weights therefore influence

the allocation of explained fit across inputs, but do not alter the fitted model itself.

Weighting changes how fit is measured, but does not affect the additivity, exactness, or model-conditional nature of the resulting attribution.

2.7 Grouped Decomposition

Because the Euler contributions sum to total explained predictive accuracy, they can be aggregated naturally across groups of components to assess group-level importance. Such grouping is useful when the number of elementary components is large, when individual contributions are noisy due to collinearity or redundancy, or when components admit meaningful economic or structural interpretations only at an aggregated level.

Suppose the prediction components are partitioned *ex ante* into disjoint groups. For a group G, define the grouped contribution as

$$C_G = \sum_{j \in G} C_j. \tag{22}$$

By additivity of the Euler decomposition,

$$\Delta\mathcal{L} = \sum_{G} C_G, \tag{23}$$

so total explained predictive accuracy is allocated exactly across groups.

This aggregation parallels the logic of Owen values (Owen, 1977), which extend Shapley allocations to pre-specified groups of features. Unlike Shapley-based approaches, however, grouped Euler contributions do not require counterfactual evaluation or refitting. Once the fitted prediction and its additive components are available, group-level contributions are obtained by direct summation at essentially no additional computational cost.

2.8 Standard Errors

Appendix A derives standard errors for the contributions to model fit. Let c_{ij} denote the observation-level contribution of input j for observation i, so that $C_j = \mathbb{E}[c_{ij}]$. An estimator of the standard error of C_j is

$$SE(C_j) = \sqrt{\frac{1}{N}\,\mathbb{E}\big[(c_{ij} - C_j)^2\big]}. \tag{24}$$

These standard errors reflect sampling variability in the data and quantify uncertainty in the estimated contributions to predictive fit across observations.

Because the attribution conditions on a fixed fitted model, the standard errors do not incorporate uncertainty about model parameter estimates. They therefore measure variability in realized contributions rather than estimation uncertainty. This perspective is appropriate for applications focused on monitoring deployed models, comparing predictive relevance across samples, or assessing changes in feature contributions over time.

Appendix A also shows how to extend these results to grouped contributions, allowing standard errors to be computed directly for aggregates of inputs without estimating a full covariance matrix.

3 Relation to Other Approaches

A surprisingly wide range of methods has been proposed for assessing feature importance in machine learning models. This diversity persists because these methods address fundamentally different questions. Clarifying these distinctions is particularly important in nonlinear regression settings, where it is often assumed that Shapley-value or perturbation-based methods are required to assess feature importance.

A useful way to distinguish these approaches is by the phase of the modeling process they are intended to support. Measures of feature importance developed in the statistics literature are primarily designed for the research and model-building phase. They help identify which variables are informative for explaining the outcome y and for selecting or refining models, often by comparing alternative specifications. Once a model has been fixed, however, these measures provide limited guidance for understanding how that model's predictive performance is generated or how it changes over time.

Many feature-importance tools developed in machine learning, by contrast, focus on explaining how features affect individual predictions. This can be valuable during the ongoing-use phase, after a model has been built. These methods are valuable for interpretability, communication, and diagnosing specific model behaviors, but they are not designed to attribute changes in overall model fit or predictive performance. Aggregating these local explanations across observations does not generally yield a stable or exact decomposition of global performance.

The Euler-style decomposition developed here addresses this gap directly. By attributing realized model fit to components of the fitted model itself, it provides a direct and computationally efficient way to understand which features drive predictive performance of an existing model and how their contributions evolve over time.

These notions of feature importance are logically distinct and should not be expected to coincide. In particular, the contribution of a feature to predictions is not the same as its contribution to model fit, and large effects on predictions may correspond to negligible or even negative contributions to predictive performance.

Figure 1 provides a graphical map that is useful for the comparisons below.

3.1 Partial R-squared

In linear regression, a classical approach to assessing feature importance is partial or incremental R^2, defined as the reduction in model fit when a regressor is removed from the model and the remaining coefficients are re-estimated. This measure is closely related to added-variable plots and dominance analysis and is sometimes interpreted as a feature's contribution to explained variance. Budescu (1993) and Draper and Smith (2014) describe this approach.

Partial R^2, however, answers a counterfactual question rather than the one considered here. Because it is defined through refitting after feature removal, it measures feature reliance or substitutability across alternative models rather than contribution to realized fit within a fixed fitted model. This perspective can be useful when exploring alternative specifications or assessing redundancy among regressors, but it does not analyze how a given fitted model generates its predictive performance.

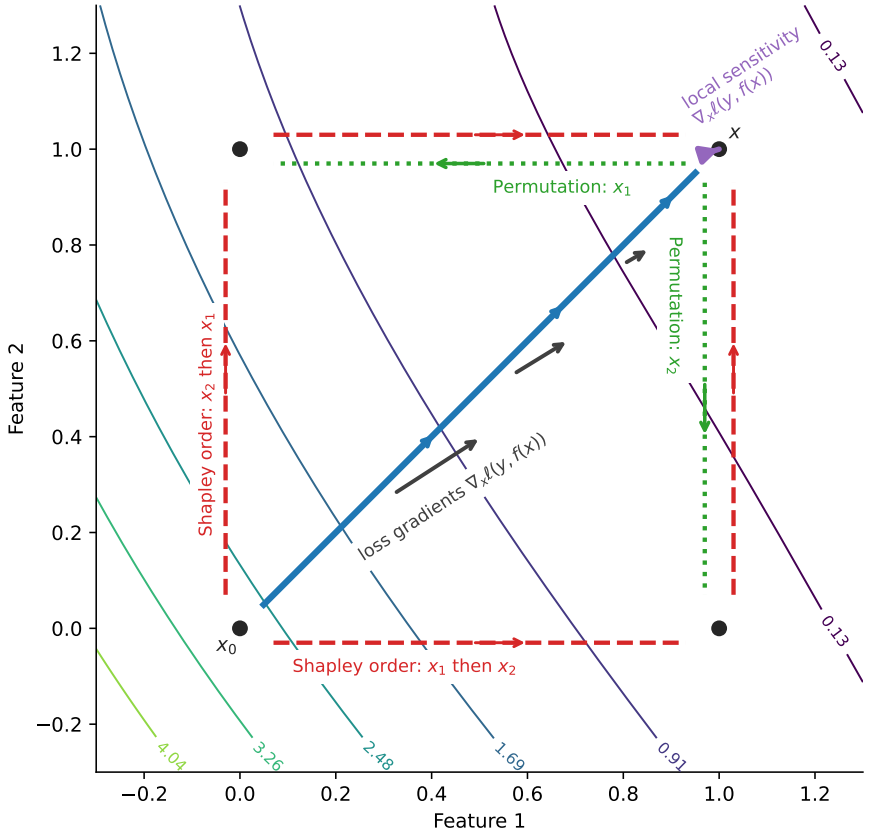
Due to refitting, partial R^2 is not additive across features: removing one regressor changes the estimated coefficients of the remaining regressors, so the associated changes in explained variance cannot be uniquely attributed or summed. When regressors are correlated, partial R^2 also depends on the order of removal when more than one regressor is removed.

The Euler-style attribution developed here can be viewed as a model-conditional analogue of variance decomposition: it attributes explained fit within the fitted model itself, yields an exact additive decomposition, and extends directly to nonlinear regression under differentiability of the loss.

3.2 Shapley Methods

Shapley-value-based methods, based on Shapley (1953), provide a very general framework for attribution based on axiomatic fairness principles. Modern implementations apply this logic to machine learning models by defining a value function over feature coalitions and averaging marginal contributions across subsets. Lindeman, Merenda, and Gold (1980), Kruskal (1987), and Lundberg and Lee (2017) describe this approach. In most practical applications, the value function is the model prediction itself, and Shapley

Figure 1: Geometry of Feature Importance Measures



The figure illustrates attribution of model fit in a bivariate nonlinear regression. All of the alternative methods are generally used to explain individual predictions, which would make comparisons inappropriate. However, the methods are quite flexible and the figure compares approaches when all methods are focused on the same measure of model fit.

The background shows level curves of the loss surface over the feature plane, with lower values indicating better fit. Explained fit is defined as the reduction in loss when moving from the baseline input $x_0 = (0, 0)$ to the realized input $x = (1, 1)$.

The solid blue diagonal line shows the straight-line path used for the Euler-style attribution developed in this paper; integrating the loss gradient along this path yields an exact, additive decomposition of explained fit across input variables for the fitted model. The illustrated loss gradients along the Euler path have a larger component in the direction of feature 1, indicating that feature 1 receives a larger share of the explained fit than feature 2. This reflects stronger alignment between feature 1 and the accumulated loss gradient along the path, which involves equal displacement in both feature directions.

The dashed red paths illustrate Shapley attribution (without refitting), which averages marginal contributions across discrete feature orderings and evaluates the loss only at corner points of the feature space rather than along a continuous path.

The dotted green paths depict permutation or perturbation methods, which remove one feature at a time starting from $x = (1, 1)$ and evaluate model fit at the adjacent nodes. These measure feature reliance or robustness rather than contribution to realized model fit and do not generally yield additive attributions.

Finally, the purple arrow at $x = (1, 1)$ shows a local gradient-based sensitivity measure $\nabla_x \ell\big(y, f(x)\big)$, which captures infinitesimal responsiveness of the loss function at x, but not contribution to overall predictive performance.

values are used to assess which features have the largest impact on individual predictions.

Conceptually, Shapley values could also be applied to a measure of model fit. In standard implementations, this would require evaluating model fit across different subsets of features. Because features are either included or excluded, with no notion of partial inclusion, this approach evaluates fit at the corners of the feature space. Interpreted in this way, Shapley values measure feature reliance: how much model performance degrades when a feature is removed.

In many applications, Shapley values are computed after refitting the model on each feature subset. In this case, the resulting attributions no longer describe performance within a fixed fitted model. Instead, they reflect feature substitutability: the extent to which other features can replace a given feature when the model is re-estimated. In this sense, however, we can interpret Shapley values applied to model fit as a combinatorial generalization of partial R^2 that averages over all feature removal paths.

The minimal assumptions imposed on the value function make Shapley methods flexible and widely applicable. However, this generality comes at the cost of substantial informational and computational inefficiency when the value function and its gradients are already well defined and this structure is not exploited.

From the perspective adopted here, Shapley-based measures address a different question from model-conditional attribution of explained fit. Their generality is valuable when the object of interest is unclear or inherently discrete, but it is unnecessary once attention is restricted to a fixed fitted model and a specific, differentiable measure of predictive performance.

3.3 Feature Perturbation Methods

Perturbation- and permutation-based importance measures are most commonly used to explain predictions rather than to attribute model fit. In their standard form, these methods assess how predictions change when inputs are corrupted, permuted, or removed, holding all other features fixed; see Breiman (2001) and Fisher, Rudin, and Dominici (2019). Averaging such effects across observations yields a global summary of how strongly each feature influences predictions.

Interpreted this way, perturbation-based importance is a measure of predictive influence or feature reliance: it quantifies how much the model's outputs depend on a given input. This notion is useful for understanding how the model forms predictions, but it does not address how predictive

accuracy is generated. Features can have a large effect on predictions while contributing little, or even negatively, to model fit.

In principle, perturbation methods are flexible enough to be applied to other objects. One could, for example, measure how a loss function changes under feature perturbations rather than how predictions change. Interpreted this way, perturbation methods assess the sensitivity of model performance to disruption of individual inputs and again yield a notion of feature reliance. For example, Gregorutti, Michel, and Saint-Pierre (2016) study variable importance in random forests by measuring changes in mean squared error under feature permutation.

Even in this formulation, however, perturbation-based approaches do not yield an additive decomposition of explained fit for a fixed fitted model. Because they rely on discrete perturbations and finite differences, the resulting importance scores do not generally sum to total explained fit and depend on the perturbation scheme, the correlation structure of the inputs, and whether the model is refit after perturbation.

It is worth noting that if all features are perturbed symmetrically and incrementally from their realized values toward a baseline input, the resulting sequence of inputs traces a straight-line path in input space similar to the Euler path used in this paper. In this sense, such path-based perturbation experiments can be viewed as sampling the loss function along the Euler path.

Perturbation methods, however, observe only changes in aggregate loss at discrete points along this path and do not provide a principled mechanism for allocating those changes across input coordinates. Recovering additive feature-level contributions from such experiments requires integrating loss gradients along the path, which is precisely the Euler-style construction introduced here.

3.4 Gradient-Based Sensitivity Methods

Gradient-based sensitivity methods are most commonly used to explain predictions rather than to attribute model fit. These approaches exploit differentiability of the prediction function and measure how predictions respond to infinitesimal changes in inputs around a point x. Belsley, Kuh, and Welsch (1980) and Hastie, Tibshirani, and Friedman (2009) describe this approach. The resulting input gradient $\nabla_x f(x)$ quantifies local predictive sensitivity and is widely used to assess which features most strongly influence individual predictions.

Interpreted this way, gradient-based measures describe responsiveness of the prediction function, not contribution to predictive accuracy. Features with

large gradients strongly affect predictions, but this alone does not indicate whether those effects improve or degrade model performance.

In principle, gradient-based methods could be applied to a loss function rather than directly to predictions. Differentiating the loss with respect to inputs yields

$$\nabla_x \ell\big(y, f(x)\big) = \ell_{\widehat{y}}\big(y, \widehat{y}\big)\ \nabla_x f(x), \tag{25}$$

where $\widehat{y} = \widehat{\alpha} + f(x)$ and $\ell_{\widehat{y}}(y, \widehat{y})$ denotes the derivative of the per-observation loss with respect to the prediction. This expression scales local prediction sensitivity by how changes in predictions translate into changes in loss.

Even in this form, however, gradient-based methods do not provide an attribution of explained fit. They measure local sensitivity at a single point x and do not aggregate effects across observations or along transitions from baseline inputs to realized inputs. Moreover, optimization of model parameters imposes no restrictions on the input gradient $\nabla_x f(x)$, even when estimation minimizes the same loss function used to evaluate performance. Stationarity conditions apply to parameters θ, not to inputs x.

As a result, local loss gradients do not generally align with contribution to overall predictive performance. A feature may exhibit large sensitivity while contributing little to explained fit if it rarely varies, varies symmetrically, or aligns weakly with the outcome. Such situations are especially common outside the training sample.

By contrast, the Euler-style attribution integrates loss gradients with respect to inputs along a path from a baseline input to the realized input, and then aggregates these effects across observations. This construction converts local sensitivities into a global, additive, and exact decomposition of a well-defined scalar measure of predictive fit.

3.5 Integrated Gradient Methods

Integrated gradient methods, described in Sundararajan, Taly, and Yan (2017), extend gradient-based sensitivity measures by integrating input gradients along a path from a baseline input to the realized input. Their primary use is to explain individual predictions by attributing the predicted value $\widehat{y}(x)$ to input features relative to a baseline.

Integrated gradients are closely related to the Euler-style attribution developed here. Both constructions rely on path integrals and exploit smoothness of the fitted prediction function. Mechanically, the two methods differ only in the object whose gradient is integrated: integrated gradients

integrate $\nabla_x f(x(t))$, the gradient of the prediction function, whereas the present framework integrates $\nabla_x \ell(y, f(x(t)))$, the gradient of a scalar loss.

This distinction determines the scope and interpretation of the resulting attributions. Integrated gradients are designed to explain individual predictions and are therefore applied observation by observation. When aggregated, they average explanations of predictions rather than decompose a global measure of model performance.

By contrast, the framework developed here targets realized predictive accuracy of a fixed fitted model, measured as the reduction in loss relative to a baseline predictor. The object being decomposed is not the prediction itself but a scalar performance functional. Applying a path-integral construction directly to the loss and then averaging across observations yields a global, additive, and exact decomposition of explained predictive fit.

This distinction also explains the different computational structure of the two approaches. Integrated gradients define feature-level contributions directly and therefore require evaluating a separate path integral for each input dimension. Euler attribution applies a path integral once to a scalar loss function; the resulting inner product decomposes algebraically into feature-level contributions. A single path integral therefore suffices regardless of the number of inputs.

From this perspective, we can view Euler attribution and integrated gradients as related path-integral methods specialized to different attribution objects. By integrating loss gradients rather than prediction gradients, the Euler-style attribution delivers a direct decomposition of realized predictive accuracy, with predictable computational cost even in high-dimensional nonlinear models.

3.6 Geometric Comparison

Figure 1 provides a geometric illustration of how the Euler-style attribution developed here differs from the feature importance methods discussed above. The comparison presumes that all methods are applied to a common object, namely model fit as measured by a scalar loss. When alternative methods are instead used to explain individual predictions, they address a different question and are not directly comparable. Although this is not the most common use of these methods, we have noted that many can be adapted to study model fit, which is the setting illustrated in the figure.

The figure shows that the Euler construction exploits information along a continuous path in the interior of the input space connecting the baseline input x_0 to the realized input x. Along this path, the attribution integrates local

loss gradients, aggregating information about how predictive performance evolves as inputs move from baseline to realization.

By contrast, the alternative methods rely on information drawn from a finite collection of points. Shapley-value methods evaluate model performance at the corners of the hypercube corresponding to feature inclusion and exclusion. Perturbation and permutation methods compare performance across collections of modified inputs that can be interpreted as boundary points obtained by corrupting, permuting, or removing individual features. Gradient-based sensitivity methods focus on local behavior at a single point x.

Even when these approaches are extended to include averaging or smoothing along paths connecting such points, their evaluations remain confined to the boundary of the input space. The Euler path integral instead traces a smooth transition through the interior of the input space. This interior path captures how loss changes continuously as multiple inputs vary jointly, which is where interaction effects naturally arise.

The figure highlights that the Euler-style attribution exploits a different and richer source of information about model performance than the leading feature importance measures. This distinction is natural and appropriate, since the Euler-style attribution is designed to answer a different and more specific question: how the predictive performance of a fixed fitted model is generated as inputs move from a baseline state to their realized values.

3.7 Computational Complexity

We have argued that the Euler-style attribution is a clean way to decompose model fit back to features. We can also show that it is computationally efficient relative to other methods that are sometimes applied to related questions. Computational efficiency is not the motivation for the construction, but it matters for practical use.

When we evaluate the Euler path integral numerically by quadrature and compute gradients by finite differences, we can state explicit function-evaluation counts. With K input features and M quadrature nodes, the attribution requires $O(MK)$ evaluations of the fitted model $f(\cdot)$. Each quadrature node requires one gradient evaluation, costing $K + 1$ model evaluations under forward differences or $2K$ under central differences. Typical choices for M are 8, 16, or 32 and are largely independent of K.[11] Importantly, the integral is taken over a one-dimensional path and does not involve numerical integration over the full K-dimensional input space. Small numerical errors in individual path integrals are likely to average out when contributions are

[11] For example, Gaussian quadrature with M nodes exactly integrates polynomials of degree up to $2M - 1$.

aggregated across observations. As a result, a full Euler-style decomposition of explained fit has predictable computational cost and remains feasible even for high-dimensional input spaces.

Direct computation of Shapley values requires $O(2^K)$ function evaluations and is infeasible except for very small K. In practice, Shapley values are approximated using Monte Carlo sampling over feature orderings or coalitions, which requires $O(PK)$ function evaluations for P samples. Common choices for P range from 10^2 to 10^4 and typically increase with K or with the degree of feature correlation in order to maintain accuracy.

In their simplest form, Perturbation and permutation methods require $O(K)$ model evaluations by perturbing each feature once. To reduce the noise inherent in a single evaluation, however, it is common to average over R Monte Carlo perturbations per feature, leading to $O(RK)$ function evaluations. Typical values of R range from 10^2 to 10^4, and larger values are often required as feature dimensionality and correlation increase in order to maintain accuracy.

Many modern machine learning systems expose the computational graph underlying the fitted prediction function $f(\widehat{\theta}, x)$ and support automatic differentiation, which computes exact derivatives by systematically applying the chain rule to this graph. (See Griewank and Walther (2008).) Using reverse-mode automatic differentiation, we can substantially reduce the computational cost of the Euler path-integral attribution.

At each quadrature node along the path, a single backward pass computes the full gradient of the loss with respect to all input features simultaneously. As a result, the total computational cost of the attribution scales as $O(M)$ forwardÐbackward evaluations of the fitted model, where M is the number of quadrature nodes, and is effectively independent of the number of features.

This contrasts sharply with Shapley- and perturbation-based methods, whose computational cost is dominated by repeated forward evaluations under modified inputs and therefore grows with the number of features K unless one is willing to accept increased approximation error in high-dimensional settings. Thus, in addition to providing an exact, model-conditional decomposition of explained fit, the Euler-style attribution achieves this decomposition at relatively low and predictable computational cost. This computational advantage becomes more pronounced in high-dimensional nonlinear models.

4 Summary

This paper develops a framework for attributing the explained predictive fit of a fixed, fitted model to input variables in nonlinear regression models. Most

existing feature-importance measures are designed to explain how features contribute to individual predictions. These are logically distinct problems: features can make large contributions to predictions without contributing to model fit. This distinction is especially important outside the training sample, where predictive performance rather than prediction sensitivity is the primary object of interest.

In linear regression, explained signal strength is a homogeneous function of the fitted values and therefore admits an exact Euler decomposition. Outside this special case, nonlinear prediction functions generally lack a canonical additive signal representation. We overcome this obstacle by applying the fundamental theorem of calculus along a path in input space, yielding an exact, additive decomposition of explained loss under mild smoothness assumptions.

Relative to related approaches, the resulting attribution evaluates the model along a continuous path through the interior of the feature space, rather than relying solely on evaluations at boundary or corner points. By accumulating marginal contributions along this path, the framework captures how predictive performance evolves as inputs move jointly from a baseline state to their realized values.

The resulting attribution is global, model-conditional, and aligned with standard measures of predictive fit such as explained variance or reduction in expected loss. It exploits structure that more general attribution methods deliberately ignore, enabling exact additivity, computational efficiency, and a clear interpretation of feature importance as contribution to realized model performance.

In addition to the contributions to model fit, we derive corresponding standard errors that reflect sampling variability in the data. These standard errors enable statistical assessment of whether observed variation in feature contributions across samples or over time is plausibly attributable to noise or instead reflects changes in predictive relevance.

By clarifying the distinction between model-conditional attribution of fit and alternative notions of feature importance based on sensitivity, perturbation, or local explanation, the framework provides a principled basis for understanding the sources of predictive performance in complex nonlinear models. Together with companion papers that provide analytical solutions for linear regression and classification models, the results establish a unified Euler-style perspective on feature importance across a broad class of predictive settings.

5 References

Belsley, David A., Edwin Kuh, and Roy E. Welsch, 1980, *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity* (Wiley, New York).

Breiman, Leo, 2001, Random forests, *Machine Learning* 45, 5–32.

Budescu, David V., 1993, Dominance analysis: A new approach to the problem of relative importance of predictors in multiple regression, *Psychological Bulletin* 114 (3), 542–551.

Draper, Norman R., and Harry Smith, 2014, *Applied Regression Analysis*, fourth edition (Wiley, Hoboken, NJ).

Fisher, Aaron, Cynthia Rudin, and Francesca Dominici, 2019, All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously, *Journal of Machine Learning Research* 20 (1), 1–81.

Gregorutti, Baptiste, Bertrand Michel, and Philippe Saint-Pierre, 2016, Correlation and variable importance in random forests, *Statistics and Computing* 27 (3), 659–678.

Griewank, Andreas, and Andrea Walther, 2008, *Evaluating Derivatives*, second edition (Society for Industrial and Applied Mathematics, Philadelphia, PA).

Guidotti, Riccardo, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi, 2018, A survey of methods for explaining black box models, *IEEE Access* 6, 61536–61556.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman, 2009, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer Series in Statistics, second edition (Springer, New York).

Hentschel, Ludger, 2026, Feature importance: Euler attribution of regression fit, Working paper, Versor Investments, New York, NY.

Kruskal, William, 1987, Relative importance by averaging over orderings, *The American Statistician* 41 (1), 6–10.

Lindeman, Richard H., Peter F. Merenda, and Ruth Z. Gold, 1980, *Introduction to Bivariate and Multivariate Analysis* (Scott, Foresman, Glenview, IL).

Lundberg, Scott M., and Su-In Lee, 2017, A unified approach to interpreting model predictions, in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 4768–4777 (Curran Associates Inc., Red Hook, NY).

Molnar, Christoph, 2022, *Interpretable Machine Learning*, second edition (Lulu.com), Available at https://christophm.github.io/interpretable-ml-book/.

Owen, Guillermo, 1977, Values of games with a priori unions, in Rudolf Henn, and Otto Moeschlin, eds., *Mathematical Economics and Game Theory*, volume 141 of *Lecture Notes in Economics and Mathematical Systems*, 76–88 (Springer, Berlin, Heidelberg).

Shapley, Lloyd S., 1953, A value for n-person games, *Contributions to the Theory of Games* 2, 307–317.

Silberberg, Eugene, 1978, *The Structure of Economics: A Mathematical Analysis* (McGraw–Hill, New York, NY).

Sundararajan, Mukund, Ankur Taly, and Qiqi Yan, 2017, Axiomatic attribution for deep networks, in Doina Precup, and Yee Whye Teh, eds., *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 3319–3328 (PMLR).

Tasche, Dirk, 2008, Capital allocation to business units and sub-portfolios: The Euler principle, in Andrea Resti, ed., *Pillar II in the New Basel Accord: The Challenge of Economic Capital*, 423–453 (Risk Books, London).

A Standard Errors

This appendix derives standard errors for the Euler contributions to explained fit in the nonlinear regression setting. The derivation parallels the linear and classification cases and requires no additional assumptions beyond those already imposed in the main text.

A.1 Observation-Level Contributions

Recall that explained fit is defined as the reduction in average loss relative to the baseline prediction,

$$\Delta\mathcal{L} = \mathcal{L}(\overline{y}) - \mathcal{L}(\widehat{y}(X)), \tag{26}$$

and that the path-integral construction yields the additive decomposition

$$\Delta\mathcal{L} = \sum_{j=1}^{K} C_j, \tag{27}$$

with component-level contributions

$$C_j = -\mathbb{E}\left[(x_j - x_{0j}) \int_0^1 \frac{\partial}{\partial x_j} \ell(y, f(x(t)))\, dt\right]. \tag{28}$$

Define the corresponding observation-level contribution for observation i as

$$c_{ij} = -(x_{ij} - x_{0j}) \int_0^1 \frac{\partial}{\partial x_j} \ell\big(y_i, f(x_i(t))\big)\, dt. \tag{29}$$

By construction,

$$C_j = \mathbb{E}[c_{ij}] = \frac{1}{N} \sum_{i=1}^{N} c_{ij}. \tag{30}$$

In practice, the integral over t is evaluated numerically using a fixed quadrature rule. This numerical approximation does not affect the asymptotic logic of the standard errors.

A.2 Standard Errors

Treating the fitted model parameters as fixed, the Euler contribution C_j is a sample average of the observation-level quantities c_{ij}. Under standard

regularity conditions,

$$\sqrt{N}\,(C_j - \mathbb{E}[c_{ij}]) \xrightarrow{d} \mathcal{N}\big(0,\, \mathbb{E}\big[(c_{ij} - C_j)^2\big]\big)\,. \tag{31}$$

Accordingly, the standard error of C_j is

$$SE(C_j) = \sqrt{\frac{1}{N}\,\mathbb{E}\big[(c_{ij} - C_j)^2\big]}. \tag{32}$$

In empirical applications, this quantity is estimated by

$$\widehat{SE}(C_j) = \sqrt{\frac{1}{N(N-1)}\sum_{i=1}^{N}(c_{ij} - C_j)^2}. \tag{33}$$

This estimator is identical in form to the standard errors used in the linear and classification settings. No modification is required for the nonlinear case.

A.3 Standard Errors and Grouped Contributions

Let $c_i = (c_{i1}, \ldots, c_{iK})^\top$ denote the vector of observation-level contributions, so that the global contribution of component j is

$$C_j = \mathbb{E}[c_{ij}], \tag{34}$$

with expectations understood as sample averages. Throughout, the fitted model is treated as fixed, and randomness arises only from sampling across observations.

Because C_j is an average of observation-level contributions, its standard error can be computed directly from the sample variance of c_{ij},

$$SE(C_j) = \sqrt{\frac{1}{N}\,\mathbb{E}\big[(c_{ij} - C_j)^2\big]} = \sqrt{\frac{1}{N(N-1)}\sum_{i=1}^{N}(c_{ij} - C_j)^2}. \tag{35}$$

This expression does not require estimation or storage of any covariance matrix and remains numerically stable even when the number of components is large.

For any group of components $G \subset \{1, \ldots, K\}$, define the observation-level group contribution

$$c_{iG} = \sum_{j \in G} c_{ij}, \tag{36}$$

and the corresponding global group contribution

$$C_G = \mathbb{E}[c_{iG}] = \sum_{j \in G} C_j. \tag{37}$$

The standard error of the grouped contribution is obtained directly from the sample variance of c_{iG},

$$SE(C_G) = \sqrt{\frac{1}{N}\,\mathbb{E}[(c_{iG} - C_G)^2]} = \sqrt{\frac{1}{N(N-1)} \sum_{i=1}^{N} (c_{iG} - C_G)^2}. \tag{38}$$

This univariate computation automatically accounts for correlation among components within the group and avoids forming any high-dimensional covariance objects.

Alternatively, we can define the covariance matrix of the vector of global contributions C as

$$\widehat{\Sigma}_C = \frac{1}{N(N-1)} \sum_{i=1}^{N} (c_i - C)(c_i - C)^\top. \tag{39}$$

Then for any group G, the grouped standard error can equivalently be written as

$$SE(C_G) = \sqrt{\mathbf{1}_G^\top \widehat{\Sigma}_C \mathbf{1}_G}, \tag{40}$$

where $\mathbf{1}_G$ is the indicator vector for the group. This expression is algebraically identical to the univariate variance formula above. In practice, however, computing $SE(C_G)$ from the observation-level group contributions c_{iG} is simpler, faster, and more robust, especially when the number of components is large.

A.4 Interpretation

These standard errors quantify sampling variability in the Euler attribution conditional on the fitted prediction function. As a result, they do not incorporate uncertainty in model estimation.

The observation-level formulation makes it straightforward to compute standard errors, confidence intervals, and group-level inference for nonlinear Euler attributions at essentially no additional computational cost once the path integrals have been evaluated.

B Decomposition Algorithm

This appendix outlines the algorithm for computing exact Euler-style contributions to explained fit in nonlinear regressions of the form $\widehat{y}(x) = \widehat{\alpha} + f(\widehat{\theta}, x)$.

The pseudo-code uses matrix notation for clarity and computational efficiency. Rows of $X \in \mathbb{R}^{N \times K}$ correspond to realized input vectors x_i, while the baseline input x_0 is a single K-vector replicated across observations. The output C_j corresponds to the contribution defined in the main text and reflects the average contribution to model fit across the full sample y.

We can compute gradients with respect to inputs via finite differences but automatic differentiation is more efficient. Under a weighted or GLS loss, we evaluate all quantities in the transformed space but attribution always is with respect to the original input coordinates.

Algorithm 1: Path-integral attribution of explained fit

```
# Notation mapping to main text:
# - X[i,:]      corresponds to x_i (realized inputs for observation i)
# - x0          corresponds to x_0 (baseline input in feature space)
# - y_pred[i]   corresponds to \widehat y(x_i) (full prediction,
    incl. intercept)
# - C[j]        corresponds to C_j
# - c[i,j]      corresponds to c_{ij} (observation-level contribution)
# - dL          corresponds to \Delta \mathcal L

# Inputs:
# y       : (N,) outcomes
# X       : (N, K) centered/standardized inputs (continuous; dummies
    OK)
# f       : prediction function; y_pred = f(X), returns (N,)
    predictions
#           IMPORTANT: f(X) returns the full fitted prediction
    \widehat y(X),
#           including any intercept \widehat\alpha.
# jac_x   : routine returning row-wise input Jacobian of the full
    prediction
#           J[i,j] = d f(X)[i] / d X[i,j], shape (N,K)
#           (typically via autodiff; finite differences as fallback)
# x0      : (K,) baseline input (default: zeros(K))
# W       : loss-metric operator (optional), defining a GLS/weighted
#           squared-error loss in transformed space.
# M       : number of quadrature nodes on [0,1] (e.g., 8, 16, 32)
```

```
# groups  : optional list of groups; each group is a list of feature
    indices
#          (e.g., groups = [G1, G2, ...], where G is a Python list of
    ints)

# Outputs:
# C       : (K,) global input contributions to explained loss
    improvement
# SE      : (K,) standard errors of C (i.i.d. sampling of
    observations)
# dL      : scalar explained loss improvement relative to baseline
    predictor f(x0)
# (optional)
# C_G     : (num_groups,) group contributions
# SE_G    : (num_groups,) group standard errors

def apply_W(Z, W):
    if W is None:
        return Z
    if is_vector(W):          # W is w_sqrt, shape (N,)
        return W * Z          # broadcasts if Z is (N,K): W[:,None] * Z
    else:                     # W is Wmat, shape (N,N)
        return W.dot(Z)       # works for (N,) and (N,K)

if x0 is None:
    x0 = zeros(K)

# Baseline inputs and baseline prediction
X0    = repeat_row(x0, N)     # (N,K)
y0_hat = f(X0)                # (N,) baseline prediction at x0

# Loss at baseline prediction (in chosen metric)
e0 = apply_W(y - y0_hat, W)  # (N,)
L0 = mean(e0 ** 2)

# Loss at realized inputs
y_hat = f(X)                  # (N,)
e     = apply_W(y - y_hat, W) # (N,)
L     = mean(e ** 2)

# Explained fit (loss improvement relative to baseline)
dL = L0 - L
```

```
# Quadrature nodes and weights on [0,1]
t_grid, a_grid = quadrature_nodes_weights_on_0_1(M)

# Accumulate observation-level contributions c_{ij}
# By construction, C_j = mean_i c_{ij}
c = zeros((N, K))

for m in range(M):
    t = t_grid[m]
    a = a_grid[m]

    Xt = X0 + t * (X - X0)    # (N,K)

    y_hat_t = f(Xt)           # (N,)
    J       = jac_x(Xt, y_hat_t) # (N,K)

    e_t    = apply_W(y - y_hat_t, W) # (N,)
    J_tilde = apply_W(J, W)       # (N,K)

    # Because dL = L0 - L, contributions enter with +2 * e_t *
        J_tilde:
    c += a * (X - X0) * (2 * e_t[:, None] * J_tilde) # (N,K)

# Global contributions
C = mean_over_i(c)            # (K,)

# Sanity check:
# sum(C) ~= dL (numerical quadrature error only)

# Standard errors, computed elementwise without any covariance matrix:
# SE(C_j) = sqrt( (1/N) * E[(c_{ij} - C_j)^2] )
# With sample averages, use the usual unbiased variance estimator:
# E[(c_{ij} - C_j)^2] = (1/(N-1)) * sum_i (c_{ij} - C_j)^2.
dc = c - C[None, :]           # (N,K)
var_c = sum_over_i(dc ** 2) / (N - 1) # (K,)
SE = sqrt(var_c / N)                # (K,)

# Optional: group contributions and group standard errors, computed
    directly.
if groups is not None:
    G = len(groups)
    C_G = zeros(G)
    SE_G = zeros(G)
```

```
    for g in range(G):
        idx = groups[g]                # list of feature indices in
            group g

        # Observation-level group contribution and its mean
        c_g = sum_over_j(c[:, idx])    # (N,)
        C_G[g] = mean(c_g)             # scalar

        # SE(C_G) = sqrt( (1/N) * E[(c_{iG} - C_G)^2] )
        dc_g = c_g - C_G[g]
        var_g = sum(dc_g ** 2) / (N - 1)
        SE_G[g] = sqrt(var_g / N)

# Notes:
# - These SEs treat the fitted model as fixed (no estimation
    uncertainty).
# - Group SEs computed from c_{iG} avoid forming any K-by-K
    covariance matrix.
```