# DO NEURAL NETWORKS PRESERVE CASE STRUCTURE? CASE-BASED DECOMPOSITION, INTERPRETATION, AND DECISION CONSISTENCY

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Neural networks increasingly influence consequential decisions, yet the case-based foundations behind these decisions remain hidden. This makes it difficult to understand which learned experiences support a decision and when such decisions can be trusted. Motivated by Case-Based Decision Theory (CBDT), which views decisions as being formed from relevant past cases, we ask whether neural networks preserve a similar case structure during learning. We first show that trained neural networks preserve a recoverable case structure within their learned representations. This structure enables a CBDT-inspired decomposition that traces fitted decisions back to contributions from individual training cases. We then reveal a fundamental gap between the recovered case structure and classical CBDT assumptions. Although neural networks preserve case-based decision structure, the recovered case coefficients do not necessarily satisfy CBDT relevance requirements. Enforcing these stricter constraints changes selected decisions and reduces task-optimal accuracy by 4.2%–9.1% relative to signed reconstruction. Finally, we establish a verifiable decision-consistency condition that identifies when the recovered case structure agrees with the original network decision. Across three real-world decision settings, decisions satisfying the condition achieve substantially higher task accuracy (83.7% vs. 35.0% on the Adult Income dataset). Case-level analysis further identifies compact sets that preserve some fitted decisions satisfying the condition. Together, our results reveal that neural networks can preserve case-based decision structures without fully satisfying CBDT assumptions. More importantly, they show that such imperfect but verifiable structures can still provide a principled foundation for reliable case-level understanding of neural network decisions.

## 1 INTRODUCTION

Neural networks learn prediction and decision functions from training cases through internal representations (Bengio et al., 2013; LeCun et al., 2015; Goodfellow et al., 2016). Prior work studies what these representations encode and how they vary across trained networks and learning (Li et al., 2016; Guo et al., 2024; Chen et al., 2024). However, these representations remain hidden during inference, which typically exposes only scores or a selected action, not which training cases support or oppose that choice or by how much. This missing connection between decisions and training cases makes the decision harder to assess and raises concerns about the reliability of these decisions.

Existing work has studied how neural-network decisions are influenced by input factors, learned representations, and training cases. At the input level, feature-attribution methods identified influential features through perturbations or additive attributions (Ribeiro et al., 2016; Lundberg & Lee, 2017; Sundararajan et al., 2017). Sanity checks further examined whether these explanations depended on the learned model and data (Adebayo et al., 2018). At the representation level, concept-based methods identified human-interpretable concepts associated with a decision (Koh et al., 2020; Xu et al., 2024; Ismail et al., 2024), while sparse-feature and circuit analyses sought latent features and causal mechanisms learned by the network itself (Huben et al., 2024; Marks et al., 2025). At the training-case level, Influence Functions and TracIn estimated how individual training examples affected model outputs through local perturbations or optimization trajectories (Koh & Liang, 2017; Pruthi et al., 2020). Related approaches studied how outputs varied across alternative training sets



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.

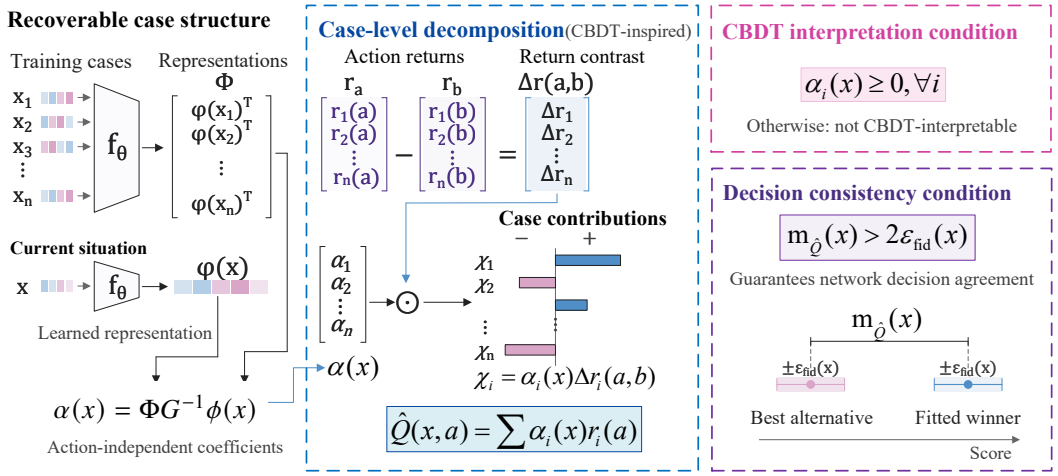



Figure 1: Overview of the case structure underlying neural network decisions. The learned representation connects neural network decisions to case-level contributions, enabling CBDT interpretation and consistency verification with the original model decisions.

or traced example effects through approximate optimization unrolling (Ilyas et al., 2022; Bae et al., 2024). Recent studies further extended data attribution to parameter-efficient language models, diffusion models, and larger computational settings (Kwon et al., 2024; Zheng et al., 2024; Schioppa, 2024). They also showed that attribution patterns changed with training stage, model scale, traced behavior, access to the training process, and parameter weighting (Wang et al., 2025; Khaddaj et al., 2025; Chang et al., 2025; Wei et al., 2025; Lee et al., 2026; Li et al., 2026). Together, these methods have made training cases visible in model analysis, while how such cases jointly form neural-network decisions has not been systematically characterized.

A case-based view offers a natural way to formulate this missing connection. CBDT evaluates each candidate action using the action-specific returns associated with relevant past cases, with more relevant cases receiving greater weight (Gilboa & Schmeidler, 1995; 2012). Case-Based Reasoning (CBR) likewise uses previous cases to reason about new problems (Kolodner, 1993; Aamodt & Plaza, 1994). In neural networks, training-point decompositions, prototype models, and surrogate methods also made individual examples visible in prediction or explanation (Yeh et al., 2018; Chen et al., 2019; Xue et al., 2024; Engel et al., 2024). Together, these lines have shown that cases can ground decisions and neural predictions can be related to examples, leaving open whether trained neural networks preserve structures that connect their decisions to the cases underlying those decisions. Here, case structure denotes the organization of case-level contributions linking a model decision to the training cases involved in learning.

In this work, we investigate whether neural networks preserve a recoverable case structure behind their decisions, when this structure admits a Case-Based Decision Theory (CBDT) interpretation, and how it can be used to assess decision reliability. Our contributions are threefold (Figure 1):

First, we show that trained neural networks preserve a recoverable case structure through their learned representations. Inspired by CBDT, this structure enables a case-level decomposition that traces fitted decisions back to contributions from individual training cases.

Second, we reveal a fundamental gap between recoverable case structure and CBDT assumptions. The recovered case coefficients do not necessarily satisfy CBDT relevance requirements, and fitting nonnegative case weights changes selected decisions and reduces task-optimal accuracy in our experiments, exposing the decision cost of this reconstruction.

Third, we establish a decision-consistency condition that identifies when the recovered case structure agrees with the original network decision. Decisions satisfying the condition achieve higher task-optimal accuracy, providing a principled approach for reliability assessment. Case-level analysis also identifies key cases that preserve some fitted decisions satisfying the condition (Appendix E.7).

## 2 Background and Problem Setup

Neural network decisions and case-based decisions provide complementary views of how decisions can be understood. This section introduces the theoretical background and problem setup used throughout the paper. We first introduce the case-based view of decision making in Section 2.1, then describe neural network decision models in Section 2.2, and finally formulate the connection between neural network decisions and case-level contributions in Section 2.3.

### 2.1 Case-Based Decision Theory

CBDT models decisions through previous cases and their associated outcomes (Gilboa & Schmeidler, 1995). It evaluates how prior cases relate to the current situation and shape preferences among alternative choices.

Let $\mathcal{D} = \{(x_i, \mathbf{r}_i)\}_{i=1}^n$ denote $n$ cases, where $x_i$ represents the observed information of case $i$, and $\mathbf{r}_i = (r_i(a))_{a\in\mathcal{A}} \in \mathbb{R}^{|\mathcal{A}|}$ collects its action-conditional returns. The finite action set $\mathcal{A}$ contains at least two actions. For each action $a$, let $\mathbf{r}^a = (r_1(a), \ldots, r_n(a))^\top \in \mathbb{R}^n$ collect its returns across cases. Given a new situation $x$, CBDT assigns each previous case a nonnegative relevance weight $\omega_i(x)$, which quantifies how relevant case $i$ is to the current situation $x$ and evaluates action $a$ as $U(x, a) = \sum_{i=1}^n \omega_i(x) r_i(a)$, where $U(x, a)$ denotes the CBDT utility assigned to action $a$ under the current situation $x$ (Gilboa & Schmeidler, 1995; 2012).

For two candidate actions $a$ and $b$, define the return difference for case $i$ as $\Delta r_i(a, b) = r_i(a) - r_i(b)$. The preference difference can then be represented as

$$\Delta U(x; a, b) = U(x, a) - U(x, b) = \sum_{i=1}^{n} \omega_i(x) \Delta r_i(a, b), \tag{1}$$

where $\Delta U(x; a, b)$ denotes the preference for action $a$ over action $b$, and $\Delta r_i(a, b)$ represents the return difference between actions $a$ and $b$ for case $i$. Equation (1) expresses decision preference as a sum of case contributions, where each contribution combines the relevance of a previous case with its relative return between competing actions. We refer to $\omega_i(x)\Delta r_i(a, b)$ as the case contribution of case $i$ to the preference between actions $a$ and $b$.

### 2.2 Neural Network Decision Models

Neural networks learn complex decision functions from large collections of training cases through parameterized representations (Bengio et al., 2013; LeCun et al., 2015). Given training cases $x_1, \ldots, x_n$ and their task-specific supervision, a neural network learns a representation function $\phi(x) = f_\theta(x) \in \mathbb{R}^H$, where $f_\theta$ denotes the neural network parameterized by $\theta$, and $\phi(x)$ denotes its learned $H$-dimensional representation. An action can be a class label or a decision with an associated return. For a set of candidate actions $\mathcal{A}$, the network assigns a score $S(x, a) = g_a(\phi(x))$ through a scoring function $g_a(\cdot)$. The final decision is $a_S(x) = \arg\max_{a\in\mathcal{A}} S(x, a)$, where $a_S(x)$ denotes the selected action. All decision rules use the same deterministic rule to resolve ties. This score function maps the current situation through a learned representation to action scores (Goodfellow et al., 2016). Although the network learns from collections of cases, its final decision does not directly reveal how individual cases contribute to the selected action.

### 2.3 Problem Setup

The two formulations above describe decision making from different perspectives. CBDT expresses decision preference through contributions from previous cases, whereas a neural network expresses it through differences between learned decision scores. Their relationship leads to the problem studied in this work. Given two candidate actions $a$ and $b$, the neural network preference is $\Delta S(x; a, b) = S(x, a) - S(x, b)$, where $\Delta S(x; a, b)$ denotes the score difference between actions $a$ and $b$ for decision instance $x$. The problem considered in this work is whether the neural network preference $\Delta S(x; a, b)$ can be expressed through a case decomposition on its learned representation, following the case-based form of the CBDT preference decomposition in Equation (1). We refer to the recoverable organization of these case-level contributions as case structure. The case coefficients $\alpha_i(x)$ introduced below are

Table 1: Formal correspondence between CBDT decision decomposition and neural network case decomposition. The table illustrates how both formulations represent decision preferences through weighted case contributions, motivating the connection studied in this work. $\widehat{Q}$ is fitted on the representation.

| Aspect | CBDT | Neural Network Case Decomposition |
|---|---|---|
| Decision score | $U(x,a) = \sum_i \omega_i(x) r_i(a)$ | $\widehat{Q}(x,a) = \sum_i \alpha_i(x) r_i(a)$ |
| Case contribution | $\omega_i(x) r_i(a)$ | $\alpha_i(x) r_i(a)$ |
| Case coefficient | $\omega_i(x) \geq 0$ | $\alpha_i(x) \in \mathbb{R}$ |
| Decision | $\arg\max_a U(x,a)$ | $\arg\max_a \widehat{Q}(x,a)$ |
| Interpretation | Case relevance under CBDT conditions | Case relevance when the conditions in Section 4 hold |

determined by the learned representation, and Section 4 will establish the conditions under which they obtain the CBDT interpretation of the nonnegative relevance weights $\omega_i(x)$ summarized in Table 1.

## 3 Case Decomposition of Neural Network Decisions

A trained neural network representation provides the basis for expressing fitted decision scores in terms of individual training cases. To recover the case structure represented by these scores, Section 3.1 first formulates action-specific decision models over the learned representation, allowing Section 3.2 to identify the action-independent case coefficients and obtain a case-level decomposition of the fitted scores. Building on this decomposition, Section 3.3 extends the case decomposition to decision margins between competing actions.

### 3.1 Action-Specific Decision Models over a Learned Representation

To obtain a case decomposition, we first fit an action-specific return model over the learned representation of the neural network. Let $\phi(x) \in \mathbb{R}^H$ denote the learned representation of the current situation $x$. For training cases $x_1, \ldots, x_n$, their representations form $\Phi = [\phi(x_1), \ldots, \phi(x_n)]^\top \in \mathbb{R}^{n \times H}$, with Gram matrix $G = \Phi^\top \Phi \in \mathbb{R}^{H \times H}$. For each candidate action $a \in \mathcal{A}$, we fit an action-specific linear return model $\widehat{Q}(x,a) = v_a^\top \phi(x)$ on the learned representation, where $v_a \in \mathbb{R}^H$ is the ordinary least squares (OLS) coefficient minimizing $\|\Phi v - \mathbf{r}^a\|_2^2$. Here, $\mathbf{r}^a = (r_1(a), \ldots, r_n(a))^\top$ collects the returns of all training cases for action $a$, where $r_i(a)$ denotes the return associated with case $i$. Assuming $\mathrm{rank}(\Phi) = H$, which requires $n \geq H$, the solution is $v_a = G^{-1}\Phi^\top \mathbf{r}^a$. Substituting this solution into the linear return model gives

$$\widehat{Q}(x,a) = \phi(x)^\top G^{-1} \Phi^\top \mathbf{r}^a. \tag{2}$$

Equation (2) separates how the current situation relates to the training cases through the learned representation from the action-specific returns. This relationship is the same across actions, while $\mathbf{r}^a$ changes with the action being evaluated. This fitted model exposes the case structure induced by a fixed learned representation. Its case terms describe fitted scores, whose connection to the original network's selected action is established in Section 5.

### 3.2 Case Decomposition of Decision Scores

To decompose a fitted decision score into training-case contributions, we define a representation-based coefficient for each case. For training case $i$, define the case coefficient as

$$\alpha_i(x) = \phi(x_i)^\top G^{-1} \phi(x). \tag{3}$$

The coefficient $\alpha_i(x)$ measures the contribution assigned to case $i$ by the learned representation when reconstructing the decision score. It depends on the representation of the current situation and determines how each training case participates in the fitted decision function. Collecting all coefficients gives $\alpha(x) = (\alpha_1(x), \ldots, \alpha_n(x))^\top = \Phi G^{-1} \phi(x)$. Each coefficient depends on the current situation and the learned representation, but not on the candidate action. Substituting the case coefficients in Eq. (3) into the fitted score in Eq. (2) gives the following result.

**Proposition 1** (Recoverable case structure and decomposition)**.** *If* $\mathrm{rank}(\Phi) = H$*, the action-independent case coefficients in Eq.* (3) *give, for every action* $a \in \mathcal{A}$*,*

$$\widehat{Q}(x, a) = \sum_{i=1}^{n} \alpha_i(x) r_i(a). \tag{4}$$

*The coefficient vector* $\alpha(x)$ *is the unique minimum-Euclidean-norm solution satisfying* $\Phi^\top \alpha(x) = \phi(x)$*.*

Equation (4) exposes the case structure defined in Section 2.3 by expressing the fitted score as a sum of training-case contributions. For case $i$, $\alpha_i(x)$ is independent of the action, while $r_i(a)$ provides the action-specific return. The same case structure therefore underlies the evaluation of different candidate actions. This decomposition follows the linear fit on fixed features and is related to training-point representations of neural predictions (Yeh et al., 2018). The same coefficients apply across actions and return vectors (Appendices A.2–A.3). Their dependence on feature coordinates is resolved below.

**Corollary 1** (Coordinate invariance)**.** *Under full column rank and OLS fitting, replacing* $\phi(x)$ *by* $B\phi(x)$ *for any invertible* $H \times H$ *matrix* $B$ *leaves* $\alpha(x)$ *unchanged. Thus the fitted scores, case-margin contributions, and decision-consistency condition are unchanged for the same network scores and returns.*

The coefficients are values of the representation kernel $K_G(x, x') = \phi(x)^\top G^{-1} \phi(x')$, with $\alpha_i(x) = K_G(x_i, x)$. The case structure is therefore independent of the chosen feature coordinates. With a fixed isotropic ridge penalty, this invariance holds for orthogonal coordinate changes (Appendix A.6).

### 3.3 Decomposition of Decision Margins

Decision selection depends on differences between the scores of candidate actions. To expose the case structure of these comparisons, we define the fitted decision margin as $\Delta\widehat{Q}(x; a, b) = \widehat{Q}(x, a) - \widehat{Q}(x, b)$. Using the case decomposition in Eq. (4), and defining $\Delta r_i(a, b) = r_i(a) - r_i(b)$, this margin can be written as

$$\Delta\widehat{Q}(x; a, b) = \sum_{i=1}^{n} \alpha_i(x) \Delta r_i(a, b). \tag{5}$$

Equation (5) shows that each case enters the fitted margin through a representation-derived coefficient $\alpha_i(x)$ and an action-dependent return contrast $\Delta r_i(a, b)$. Their product defines the case-level contribution to that margin. Accordingly, define the case contribution as $\chi_i(x; a, b) = \alpha_i(x) \Delta r_i(a, b)$. The fitted margin is then $\Delta\widehat{Q}(x; a, b) = \sum_{i=1}^{n} \chi_i(x; a, b)$. The sign and magnitude of $\chi_i(x; a, b)$ show whether and how strongly case $i$ contributes to the fitted preference between actions $a$ and $b$. A positive contribution supports $a$ over $b$, whereas a negative contribution supports $b$ over $a$. Equation (4) decomposes fitted scores, and Eq. (5) decomposes their pairwise margins. The coefficients themselves can also be negative, raising the question of whether they admit the CBDT meaning of case relevance.

## 4 Case-Based Interpretation of Neural Network Decisions

The recovered case structure reveals how individual cases contribute to fitted decisions, but it does not automatically imply a CBDT interpretation. This section characterizes the nonnegative relevance condition required by CBDT and examines how enforcing this condition changes the recovered decision structure.

### 4.1 Conditions for Case-Based Interpretation

The decomposition in Section 3 yields case-level coefficients, but these coefficients admit a Case-Based Decision Theory interpretation only when they satisfy the structural requirements of CBDT. In CBDT, decisions are formulated by combining the relevance of previous cases with their associated action returns. Specifically, the value of action $a$ is represented as $U(x, a) = \sum_{i=1}^{n} \omega_i(x) r_i(a)$ (Gilboa & Schmeidler, 1995; 2012). The fitted decision score obtained from the neural network

representation follows the same case-return structure: $\widehat{Q}(x,a) = \sum_{i=1}^{n} \alpha_i(x) r_i(a)$, where $\alpha_i(x)$ denotes the coefficient associated with case $i$. The two formulations differ in how these coefficients are obtained and interpreted. In CBDT, relevance weights $\omega_i(x)$ are nonnegative because they represent the relevance of previous cases when evaluating an action (Gilboa & Schmeidler, 1995). In contrast, the neural network coefficients $\alpha_i(x)$ are determined by the learned representation and do not automatically satisfy the nonnegativity condition required for a CBDT interpretation. Therefore, the case coefficients admit a case-based interpretation when

$$\alpha_i(x) \geq 0, \qquad i = 1, \ldots, n. \tag{6}$$

Under this condition, $\alpha_i(x)$ can be interpreted as the CBDT relevance assigned to case $i$. The recovered case structure therefore admits a Case-Based Decision Theory interpretation in which each case contributes according to its relevance and associated return. When this condition is not satisfied, the decomposition remains mathematically valid, but its coefficients cannot serve as CBDT relevance weights. Thus, case decomposition and Case-Based Decision Theory interpretation are distinct properties of the recovered case structure. We also construct nonnegative weights $\beta(x) \in \arg\min_{u \geq 0} \|\Phi^\top u - \phi(x)\|_2^2$ by nonnegative least squares. This fits the query representation using nonnegative case weights, whose induced scores can differ from $\widehat{Q}$. The minimizing weights need not be unique, so the reported scores use the vector returned by the solver. Further analyses of clipping error bounds and reconstruction and score deviations are provided in Appendices A.4 and D.4.

### 4.2 Decision Interpretation through Case Contributions

To interpret how individual cases enter a decision preference, we examine their contributions within a pairwise action comparison. For two candidate actions $a$ and $b$, the case contribution to the fitted decision margin is defined as $\chi_i(x;a,b) = \alpha_i(x)\Delta r_i(a,b)$. Under this interpretation condition, the coefficients $\alpha_i(x)$ can be assigned the CBDT meaning of relevance weights, allowing the recovered case contributions to represent supporting and opposing cases in the decision process. The fitted decision margin is therefore the sum of individual case contributions, $\Delta\widehat{Q}(x;a,b) = \sum_{i=1}^{n} \chi_i(x;a,b)$. A positive $\chi_i(x;a,b)$ supports $a$ over $b$, whereas a negative value supports $b$ over $a$. Under the interpretation condition, these contributions provide a case-based account of the fitted preference through its supporting and opposing cases.

## 5 Decision Consistency of Case-Based Interpretations

Although the recovered coefficients may be signed, the resulting case structure can still explain the decision selected by the original network. This section establishes decision consistency between the recovered decomposition and the network decision, and uses this condition to assess empirical decision reliability.

### 5.1 Decision Consistency Criterion

The recovered case structure is obtained from the decomposed decision formulation $\widehat{Q}(x,a)$, while the original neural network produces decisions through its score function $S(x,a)$. Its fitted decision agrees with the original neural network when the fitted formulation and the neural network select the same action. For a decision instance $x$, the consistency requirement is defined as

$$a_{\widehat{Q}}(x) = \arg\max_{a \in \mathcal{A}} \widehat{Q}(x,a) = \arg\max_{a \in \mathcal{A}} S(x,a) = a_S(x). \tag{7}$$

where $a_{\widehat{Q}}(x)$ denotes the action selected by the fitted formulation and $a_S(x)$ denotes the action selected by the original neural network. Decision consistency establishes agreement on the selected action, but it does not require the fitted and network margins to have identical magnitudes. Separately, the CBDT interpretation condition determines whether the recovered coefficients admit a case-based relevance interpretation.

### 5.2 Sufficient Conditions for Decision Consistency

Decision consistency depends on whether the discrepancy between the fitted and neural network scores can change the selected action. Let the deviation between the original neural network score

and the fitted decision score be bounded by

$$|S(x,a) - \widehat{Q}(x,a)| \le \varepsilon_{\text{fid}}(x), \qquad a \in \mathcal{A}. \tag{8}$$

where $\varepsilon_{\text{fid}}(x)$ denotes the maximum deviation between the original neural network score and the fitted decision score for decision instance $x$. The following result links the fitted winning margin to decision consistency.

**Theorem 1** (Decision consistency)**.** *Fix a decision instance $x$ and suppose Eq.* (8) *holds for every action. If the winning margin of the fitted formulation satisfies*

$$\widehat{Q}(x, a_{\widehat{Q}}(x)) - \max_{b \neq a_{\widehat{Q}}(x)} \widehat{Q}(x,b) > 2\varepsilon_{\text{fid}}(x), \tag{9}$$

*then $a_{\widehat{Q}}(x) = a_S(x)$.*

Under Eq. (9), the score error cannot reverse the ordering between the fitted winner and any competitor, so the fitted formulation and the original neural network select the same action. The factor two is tight: opposite score errors can close a fitted margin of $2\varepsilon_{\text{fid}}(x)$ and reverse any smaller positive margin (Appendix A.5). Theorem 1 guarantees fitted–network agreement for both signed and nonnegative case coefficients, while the CBDT condition separately determines their relevance interpretation. Ranking decision instances by the fitted winning margin relative to twice the score error provides a decision-selection rule. The relationship between this margin-to-error selection and empirical decision reliability is evaluated in Section 6.3.

## 6 Experiments

We evaluate recoverable case structure in trained neural networks using a controlled CBDT setting and three decision tasks based on real-world data. The controlled setting provides explicit cases and decision outcomes for evaluating case decomposition (Section 6.1). The Pennsylvania–New Jersey–Maryland (PJM) bidding task (PJM Interconnection, 2017), Adult Income (Becker & Kohavi, 1996), and Default of Credit Card Clients (Yeh, 2009; Yeh & Lien, 2009) evaluate the decision effects of nonnegative reconstruction and the reliability of decisions satisfying the condition (Sections 6.2–6.3). Task construction, network configurations, and evaluation measures are specified in Appendix B.

### 6.1 Case Decomposition Results

We first evaluate whether the learned representation preserves a recoverable case structure by expressing fitted decision scores through individual case contributions.

On the controlled CBDT setting, the neural network closely matches the generating CBDT decision regions (Figure 2(a,b)), while fitted scores recover the underlying case support (Figure 2(c)). The

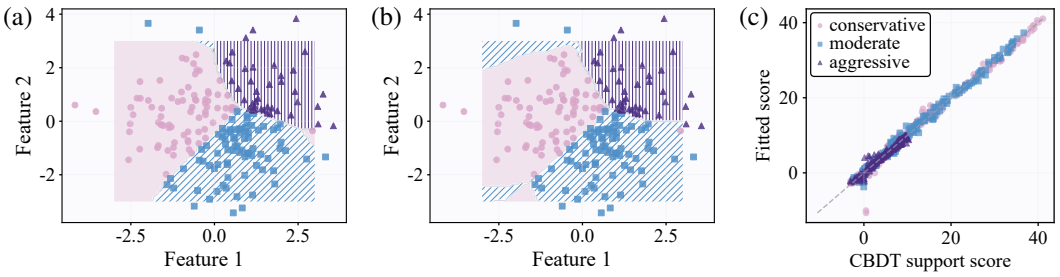



Figure 2: Recovery of case structure from neural network representations on the controlled CBDT setting. (a) Ground-truth CBDT decision structure generated from the underlying case-based process. (b) Decision structure produced by the trained neural network over the learned representation. The network decision regions closely align with those of the reference structure. (c) Correspondence between recovered and reference support scores, where the diagonal relationship indicates recovery of aggregate support.

Table 2: Case contribution correspondence with the reference structure on the controlled CBDT setting. Ranking is evaluated by Top-30 Jaccard and normalized discounted cumulative gain (nDCG); sign agreement measures support direction. The decomposition aligns more closely with the specified reference than the comparison scores.

| Scoring rule | Pearson ↑ | Top-30 Jaccard ↑ | Top-30 nDCG ↑ | Top-30 sign agreement ↑ |
|---|---|---|---|---|
| **Case decomposition** | **0.600** | **0.390** | **0.711** | **0.828** |
| Influence Functions (linear model) | −0.023 | 0.124 | 0.360 | 0.498 |
| Representer-style (linear model) | 0.026 | 0.043 | 0.166 | 0.395 |
| Single-checkpoint gradient / inner product | −0.026 | 0.043 | 0.166 | 0.605 |

recovered case contributions also align with the specified relevance-and-return reference over training cases (Table 2), linking the learned decision preferences to their case-level structure. The representation ablation further examines the role of the learned representation: the trained representation reconstructs neural network scores with $R^2 = 1.000$, produces support correlation of 0.996, and satisfies the consistency condition for 94.3% of evaluated inputs, while replacing it with random representations substantially reduces recovery consistency. These results indicate that the recoverable case structure is supported by information encoded in the learned representation. Full ablation and checkpoint results are reported in Appendix C, with comparison details and costs in Appendix F.

### 6.2 Case-Based Interpretation Results

We next examine whether the case decomposition obtained from neural network decisions admits a Case-Based Decision Theory interpretation. On the PJM bidding task, neural network and CBDT decisions agree closely despite different underlying case-support patterns (Figure 3). Although the selected actions are consistent, the recovered case structures are not equivalent: all 250 examined coefficient vectors contain negative entries, preventing a direct CBDT interpretation. This shows that decision agreement alone does not determine how individual cases support a choice.

We further evaluate whether enforcing CBDT-style nonnegative relevance preserves the recovered decisions. Across the three real-world decision tasks, the nonnegative fits closely reconstruct the learned representations, with mean relative representation residuals of 0.0006–0.0019. However, this representation-level agreement does not preserve the original decisions: fitted-score deviations reach 0.20–0.42, causing 14.7%–28.1% of selected actions to change. A separate evaluation finds relative decreases in task-optimal accuracy of 4.2%, 6.3%, and 9.1% on Adult, Default, and bidding, respectively. These results show that representation reconstruction alone is insufficient to preserve

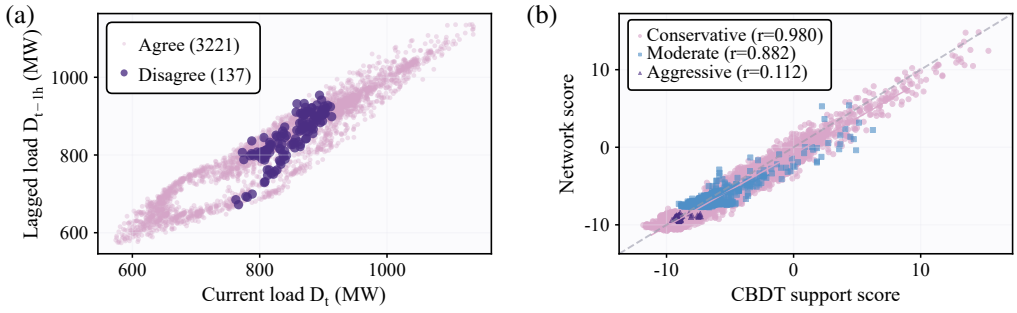
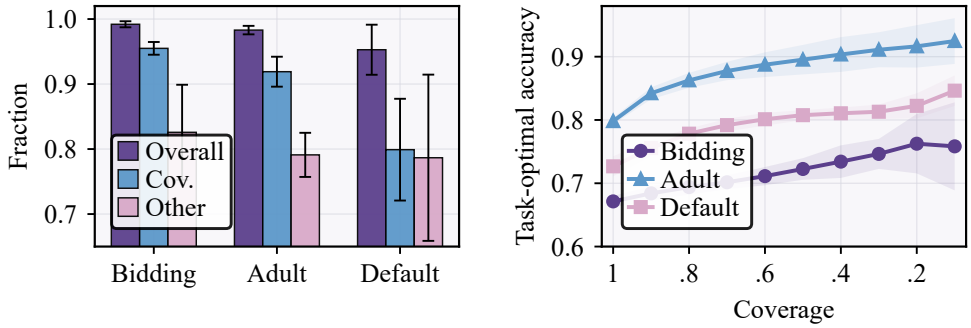



Figure 3: Decision agreement and case support correspondence on the PJM bidding task. (a) Agreement between neural network and CBDT selected actions, where the two models select the same bidding action for 95.9% of inputs; loads are in megawatts (MW). (b) Conservative, Moderate, and Aggressive denote the three bidding strategies evaluated in the task. Correspondence between neural network scores and explicit CBDT support, where the Aggressive bidding action achieves only 0.112 support-score correlation despite action agreement.




(a) Decision consistency (b) Accuracy at retained coverage

Figure 4: Decision consistency and reliability selection using the recovered case structure. (a) Condition-based separation of decisions. Overall denotes fitted–network decision agreement, Cov. denotes consistency coverage, and Other denotes agreement among remaining inputs. Across three real-world tasks, overall fitted–network agreement remains high, yet the condition separates inputs with guaranteed agreement from the remaining ones. (b) Selective reliability using the margin-to-error ratio. Ranking inputs by this ratio identifies subsets with different task-optimal accuracy. Higher-ranked inputs achieve higher accuracy under reduced coverage on Adult and Default, while bidding peaks at 20% coverage. Bars and curves are five-seed means with 95% intervals.

decision quality, because the recovered decision also depends on case-return contributions. Detailed score identities and reconstruction analyses are provided in Appendix D.4.

### 6.3 Decision Consistency Results

We finally examine whether the recovered case structure agrees with the network decision and whether the consistency condition identifies decisions with greater empirical reliability. Across the three real-world decision tasks (PJM bidding, Adult Income, and Default of Credit Card Clients), fitted and network decisions largely agree (Figure 4(a)). Inputs satisfying the condition achieve 76.4% task-optimal accuracy on average, compared with 46.6% for remaining inputs. The largest separation occurs on the Adult Income task, where inputs satisfying the condition reach 83.7% accuracy compared with 35.0% for remaining inputs. This gap shows that the consistency condition provides a practical criterion for identifying decisions whose recovered case structure is more likely to correspond to reliable outcomes.

We further extend this binary consistency condition to different coverage levels by ranking inputs according to the fitted margin-to-error ratio (Figure 4(b)). Retaining the highest-ranked 10% of inputs increases average task-optimal accuracy from 73.2% to 84.3%, corresponding to an improvement of 11.1 percentage points. The largest gain occurs on Adult Income, where accuracy increases from 79.8% to 92.5% (12.7 percentage points). These averages weight the three real-world tasks equally. The selection rule relies only on network and fitted scores available at inference time. Observed returns are used only for evaluating task accuracy on held-out inputs. Full selection results and additional case-level analyses are provided in Appendix E.

## 7 Conclusion

Neural networks preserve a recoverable case structure that traces fitted decisions back to contributions from training cases. The recovered case coefficients may be signed, and enforcing nonnegative case relevance can alter decisions even when the learned representations are nearly unchanged. Our consistency condition identifies when the recovered structure agrees with the network's selected action, and the margin-to-error ratio provides a practical rule for selecting decisions with higher empirical accuracy across most coverage levels. Future work will further investigate the rationality of neural network decisions through decision-theoretic criteria.

### AI USE STATEMENT

AI tools assisted with polishing manuscript language, language translation, searching for related literature, and document checks. The authors manually verified the literature identified through AI-assisted search. All other research content and activities, including the research ideas, method development, theoretical derivations, experimental design and execution, and analysis of the results, were completed by the authors. The authors take responsibility for the content of this paper.

### ETHICS STATEMENT

This study uses public PJM load data, Adult Income, and Default of Credit Card Clients in computational experiments. It involves no recruitment of participants or animal experimentation. The action returns are constructed for methodological evaluation. Adult and Default contain demographic and financial attributes, so case-level traceability does not establish fairness or justify decisions affecting individuals. The analysis uses the released attributes without attempting to identify individuals.

### REPRODUCIBILITY STATEMENT

The assumptions and proofs are given in Appendix A. Task construction, data processing, network configurations, fitting procedures, and seed-level evaluation are described in Appendix B. Appendices C–F provide the additional analyses and computational costs. Experiment scripts and recorded outputs are supplied as an anonymous supplementary code attachment. The three data sources are cited in Appendix B, and the PJM experiments use the processed half-hourly series described there.

## REFERENCES

Agnar Aamodt and Enric Plaza. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications*, 7(1):39–59, 1994. doi: 10.3233/AIC-1994-7104. URL https://doi.org/10.3233/AIC-1994-7104.

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity Checks for Saliency Maps. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/294a8ed24b1ad22ec2e7efea049b8737-Paper.pdf.

Juhan Bae, Wu Lin, Jonathan Lorraine, and Roger Grosse. Training Data Attribution via Approximate Unrolling. In *Advances in Neural Information Processing Systems*, volume 37, pp. 66647–66686. Curran Associates, Inc., 2024. doi: 10.52202/079017-2129. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/7af60ccb99c7a434a0d9d9c1fb00ca94-Paper-Conference.pdf.

Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. URL https://archive.ics.uci.edu/dataset/2/adult. Dataset. Accessed March 22, 2026.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013. doi: 10.1109/TPAMI.2013.50. URL https://doi.org/10.1109/TPAMI.2013.50.

Tyler Chang, Dheeraj Rajagopal, Tolga Bolukbasi, Lucas Dixon, and Ian Tenney. Scalable Influence and Fact Tracing for Large Language Model Pretraining. In *International Conference on Learning Representations*, pp. 40976–40997, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/65798a76cc176c29b6bfefe84b0a03ff-Paper-Conference.pdf.

Angelica Chen, Ravid Shwartz-Ziv, Kyunghyun Cho, Matthew Leavitt, and Naomi Saphra. Sudden Drops in the Loss: Syntax Acquisition, Phase Transitions, and Simplicity Bias in MLMs. In *International Conference on Learning Representations*, pp. 14479–14510, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/3e8df255a03b383f30abe09203770213-Paper-Conference.pdf.

Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This Looks Like That: Deep Learning for Interpretable Image Recognition. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/adf7ee2dcf142b0e11888e72b43fcb75-Paper.pdf.

Andrew Engel, Zhichao Wang, Natalie Frank, Ioana Dumitriu, Sutanay Choudhury, Anand Sarwate, and Tony Chiang. Faithful and Efficient Explanations for Neural Networks via Neural Tangent Kernel Surrogate Models. In *International Conference on Learning Representations*, pp. 27239–27292, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/7319b7561ffe5e2f6419acd4a2f52d6b-Paper-Conference.pdf.

Itzhak Gilboa and David Schmeidler. Case-Based Decision Theory. *The Quarterly Journal of Economics*, 110(3):605–639, 1995. doi: 10.2307/2946694. URL https://academic.oup.com/qje/article-abstract/110/3/605/1859208.

Itzhak Gilboa and David Schmeidler. *Case-Based Predictions: An Axiomatic Approach to Prediction, Classification and Statistical Learning*. World Scientific, 2012. doi: 10.1142/8243. URL https://doi.org/10.1142/8243.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. ISBN 9780262035613. URL https://www.deeplearningbook.org/.

Tianyu Guo, Wei Hu, Song Mei, Huan Wang, Caiming Xiong, Silvio Savarese, and Yu Bai. How Do Transformers Learn In-Context Beyond Simple Functions? A Case Study on Learning with Representations. In *International Conference on Learning Representations*, pp. 46014–46042, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/c9694bf4f9bf3626f7d21158bab74f8e-Paper-Conference.pdf.

Robert Huben, Hoagy Cunningham, Logan Smith, Aidan Ewart, and Lee Sharkey. Sparse Autoencoders Find Highly Interpretable Features in Language Models. In *International Conference on Learning Representations*, pp. 7827–7845, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/1fa1ab11f4bd5f94b2ec20e794dbfa3b-Paper-Conference.pdf.

Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Understanding Predictions with Data and Data with Predictions. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9525–9587. PMLR, 2022. URL https://proceedings.mlr.press/v162/ilyas22a.html.

Aya Abdelsalam Ismail, Julius Adebayo, Hector Corrada Bravo, Stephen Ra, and Kyunghyun Cho. Concept Bottleneck Generative Models. In *International Conference on Learning Representations*, pp. 34114–34132, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/9149fc44c95ce58e3ca529a1e34c2691-Paper-Conference.pdf.

Alaa Khaddaj, Logan Engstrom, and Aleksander Madry. Small-to-Large Generalization: Training Data Influences Models Consistently Across Scale. In *International Conference on Learning Representations*, pp. 33729–33768, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/5356603f9c47399adfd372f77a677057-Paper-Conference.pdf.

Pang Wei Koh and Percy Liang. Understanding Black-box Predictions via Influence Functions. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1885–1894. PMLR, 2017. URL https://proceedings.mlr.press/v70/koh17a.html.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept Bottleneck Models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5338–5348. PMLR, 2020. URL https://proceedings.mlr.press/v119/koh20a.html.

Janet L. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann, 1993. ISBN 9781558602373. doi: 10.1016/C2009-0-27670-7. URL https://www.sciencedirect.com/book/9781558602373/case-based-reasoning.

Yongchan Kwon, Eric Wu, Kevin Wu, and James Y Zou. DataInf: Efficiently Estimating Data Influence in LoRA-tuned LLMs and Diffusion Models. In *International Conference on Learning Representations*, pp. 21921–21942, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/5e84a0f233611a1dc8fb794dc52415a3-Paper-Conference.pdf.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep Learning. *Nature*, 521(7553):436–444, 2015. doi: 10.1038/nature14539. URL https://www.nature.com/articles/nature14539.

Jin Hwa Lee, Matthew Smith, Maxwell Adam, and Jesse Hoogland. Influence Dynamics and Stagewise Data Attribution. In *International Conference on Learning Representations*, pp. 35716–35747, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/3c1edb917edb6cec59bf3a0afc03150e-Paper-Conference.pdf.

Shuangqi Li, Hieu Le, Jingyi Xu, and Mathieu Salzmann. Learning to Weight Parameters for Training Data Attribution. In *International Conference on Learning Representations*, pp. 70540–70570, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/72fb9ab442fc60b7ae5d53bf6b478273-Paper-Conference.pdf.

Yixuan Li, Jason Yosinski, Jeff Clune, Hod Lipson, and John Hopcroft. Convergent Learning: Do Different Neural Networks Learn the Same Representations? In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1511.07543.

Scott M Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.

Samuel Marks, Can Rager, Eric Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse Feature Circuits: Discovering and Editing Interpretable Causal Graphs in Language Models. In *International Conference on Learning Representations*, pp. 23888–23923, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/3ba4d47a83e498c2b1a0868cba20f6de-Paper-Conference.pdf.

PJM Interconnection. Historical Metered Load Data. PJM Data Directory, 2017. URL https://www.pjm.com/markets-and-operations/data-dictionary. 2017 load data. Accessed July 12, 2023.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating Training Data Influence by Tracing Gradient Descent. In *Advances in Neural Information Processing Systems*, volume 33, pp. 19920–19930. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/e6385d39ec9394f2f3a354d9d2b88eec-Paper.pdf.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144. ACM, 2016. doi: 10.1145/2939672.2939778. URL https://doi.org/10.1145/2939672.2939778.

Andrea Schioppa. Efficient Sketches for Training Data Attribution and Studying the Loss Landscape. In *Advances in Neural Information Processing Systems*, volume 37, pp. 37692–37735. Curran Associates, Inc., 2024. doi: 10.52202/079017-1190. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/4268cf6ba253744befa41ffee5083fa0-Paper-Conference.pdf.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic Attribution for Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3319–3328. PMLR, 2017. URL https://proceedings.mlr.press/v70/sundararajan17a.html.

Jiachen (Tianhao) Wang, Dawn Song, James Y Zou, Prateek Mittal, and Ruoxi Jia. Capturing the Temporal Dependence of Training Data Influence. In *International Conference on Learning Representations*, pp. 11236–11273, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/1e0d38c676d5855bcfab7f6d29d20ad9-Paper-Conference.pdf.

Dennis Wei, Inkit Padhi, Soumya Ghosh, Amit Dhurandhar, Karthikeyan Natesan Ramamurthy, and Maria Chang. Final-Model-Only Data Attribution with a Unifying View of Gradient-Based Methods. In *Advances in Neural Information Processing Systems*, volume 38, pp. 106802–106836. Curran Associates, Inc., 2025. doi: 10.52202/085713-3562. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/99d7326032bbed26de1b244beaff6a84-Paper-Conference.pdf.

Xinyue Xu, Yi Qin, Lu Mi, Hao Wang, and Xiaomeng Li. Energy-Based Concept Bottleneck Models: Unifying Prediction, Concept Intervention, and Probabilistic Interpretations. In *International Conference on Learning Representations*, pp. 6944–6966, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/1c364d98a5cdc426fd8c76fbb2c10e34-Paper-Conference.pdf.

Mengqi Xue, Qihan Huang, Haofei Zhang, Jingwen Hu, Jie Song, Mingli Song, and Canghong Jin. ProtoPFormer: Concentrating on Prototypical Parts in Vision Transformers for Interpretable Image Recognition. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 1516–1524, 2024. doi: 10.24963/ijcai.2024/168. URL https://www.ijcai.org/proceedings/2024/168.

Chih-Kuan Yeh, Joon Kim, Ian En-Hsu Yen, and Pradeep K Ravikumar. Representer Point Selection for Explaining Deep Neural Networks. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/8a7129b8f3edd95b7d969dfc2c8e9d9d-Paper.pdf.

I-Cheng Yeh. Default of Credit Card Clients. UCI Machine Learning Repository, 2009. URL https://archive.ics.uci.edu/dataset/350/default+of+credit+card+clients. Dataset. Accessed March 22, 2026.

I-Cheng Yeh and Che-hui Lien. The Comparisons of Data Mining Techniques for the Predictive Accuracy of Probability of Default of Credit Card Clients. *Expert Systems with Applications*, 36(2): 2473–2480, 2009. doi: 10.1016/j.eswa.2007.12.020. URL https://www.sciencedirect.com/science/article/pii/S0957417407006719.

Xiaosen Zheng, Tianyu Pang, Chao Du, Jing Jiang, and Min Lin. Intriguing Properties of Data Attribution on Diffusion Models. In *International Conference on Learning Representations*, pp. 18417–18452, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/50be7e77b9c883144940be925b608acc-Paper-Conference.pdf.

# A MATHEMATICAL DETAILS

Table A1: Notation used in the case decomposition, CBDT interpretation condition, decision-consistency bound, and experimental evaluations.

| Symbol | Meaning | Symbol | Meaning |
|---|---|---|---|
| $\mathcal{X}, \mathcal{A}$ | Situation space and finite action set | $\mathcal{D}$ | Collection of training cases and returns |
| $x,\ x_i$ | Current situation and training case $i$ | $n,\ H$ | Number of training cases; representation dimension |
| $a,\ b$ | Two candidate actions | $\mathbf{r}_i$ | Returns for case $i$; dimension $\lvert\mathcal{A}\rvert$ |
| $r_i(a)$ | Return of action $a$ for case $i$ | $\mathbf{r}^a$ | Returns across $n$ cases for action $a$ |
| $\phi(x)$ | Column representation; dimension $H$ | $\Phi$ | Training representation matrix; $n \times H$ |
| $G$ | Gram matrix $\Phi^\top\Phi$; $H \times H$ | $I_H$ | Identity matrix of dimension $H$ |
| $S(x, a)$ | Neural network score for action $a$ | $\widehat{Q}(x, a)$ | Fitted score; OLS in Section 3, chosen fit in Sections 4–5 |
| $v_a$ | OLS weight vector for action $a$ | $\alpha_i(x)$ | Coefficient of case $i$ in the chosen fit |
| $\alpha(x)$ | Action-independent coefficient vector; dimension $n$ | $\lambda$ | Ridge regularization strength |
| $\alpha_\lambda(x)$ | Ridge vector; entry $\alpha_{\lambda,i}(x)$ | $\widehat{Q}_\lambda$ | Ridge-fitted return score |
| $\Delta r_i(a, b)$ | Case return contrast: $r_i(a) - r_i(b)$ | $\chi_i(x; a, b)$ | Pairwise case contribution; $\chi_{\lambda,i}$ for ridge |
| $\Delta\widehat{Q}$ | Fitted score difference between actions | $\omega_i(x)$ | Nonnegative CBDT case relevance |
| $U(x, a)$ | CBDT support for action $a$ | $a_S(x)$ | Action selected by the neural network |
| $a_{\widehat{Q}}(x)$ | Action selected by the fitted model | $m_{\widehat{Q}}(x)$ | Fitted winning margin |
| $\varepsilon_{\mathrm{fid}}(x)$ | Maximum fitted–network score error | $\alpha_i^+(x), \alpha_i^-(x)$ | Positive and negative coefficient parts |
| $\nu(x), \nu_\lambda(x)$ | Negative mass for the chosen fit and ridge | $R_a$ | Maximum absolute return for action $a$ |
| $Q^+(x, a)$ | Score after clipping coefficients at zero | $k,\ \Phi_k$ | Checkpoint index or case-prefix length, and checkpoint representation matrix |
| $G_k, \alpha_{k,\lambda}$ | Checkpoint Gram matrix and ridge coefficients | $B,\ G_B$ | Invertible coordinate map and transformed Gram matrix |
| $a_{\mathrm{gt}}, b_{\mathrm{gt}}$ | Generator-selected action and runner-up | $\chi_i^{\mathrm{gt}}$ | Reference contribution on sampled training cases |
| $\sigma_r, \sigma_a, \eta_{i,a}$ | Noise level; return standard deviation; Gaussian draw | $R^2$, error | Coefficient of determination; mean squared error (MSE) |
| Jaccard, nDCG | Case-set overlap and normalized discounted cumulative gain | Uncertainty | Standard deviation (SD); confidence interval (CI) |
| $f_\theta, g_a$ | Representation map and action scoring function | $\theta$ | Neural network parameters |
| $v_{\lambda,a}$ | Ridge weight vector for action $a$ | $\Delta S, \Delta U$ | Network and CBDT score differences |

| Symbol | Meaning | Symbol | Meaning |
|---|---|---|---|
| $\phi_k, \widehat{Q}_{k,\lambda}$ | Checkpoint representation and fitted score | $\widehat{a}$ | Shorthand for $a_{\widehat{Q}}(x)$ |
| $N, \bar{S}_a$ | Evaluation count and mean network action score | $\tilde{r}_i(a), \bar{r}_a$ | Perturbed return and mean training return |
| $K_i(x), e_i(a,b)$ | Feature inner product and residual contrast | $H_d, \lambda_{\mathrm{rep}}$ | Damped comparison matrix and representer scale |
| $E_\lambda^{\mathrm{avg}}$ | Action-averaged clipping bound | $A_{\mathrm{all}}, A_{\mathrm{unc}}$ | Overall and remaining-input agreement rates |
| $K_G$ | OLS representation kernel | $\beta, Q_\beta$ | Nonnegative reconstruction weights and induced score |
| $P_\Phi, P_{\Phi,\lambda}$ | OLS and ridge fitted-value matrices | $R, V_\lambda, E_\lambda, d_\beta$ | Return matrix, ridge parameters, training residuals, and query reconstruction residual |
| $\rho, c, t$ | Margin-to-error ratio, retained coverage, and clipping fraction | $u$ | Candidate case-coefficient vector |

Sections 3–5 address three properties of the recovered case structure: how a learned representation yields a case decomposition, when the recovered coefficients admit a case-based interpretation, and when the resulting case-based formulation remains consistent with the original neural network decision. The notation is summarized in Table A1, followed by the proofs in that order. Experimental settings are specified in Appendix B, with corresponding evidence in Appendices C–E and complementary comparisons and computational costs in Appendix F.

### A.1 Notation and assumptions

Let $\mathcal{D} = \{(x_i, \mathbf{r}_i)\}_{i=1}^n$ be the training cases, with $\mathbf{r}_i \in \mathbb{R}^{|\mathcal{A}|}$ and $\mathbf{r}^a \in \mathbb{R}^n$ as defined in Section 2.1. The action set is finite and contains at least two actions. A common deterministic rule resolves ties. The query $x \in \mathcal{X}$ may be a training situation or a new situation, where $\mathcal{X}$ denotes the situation space. Returns are fixed when fitting the linear return models; the experiments construct a return for every action and case.

The neural network supplies scores $S(x,a)$ and a learned representation $\phi(x) \in \mathbb{R}^H$. During case-structure recovery, this learned representation is treated as fixed. Row $i$ of $\Phi \in \mathbb{R}^{n\times H}$ is $\phi(x_i)^\top$, and $G = \Phi^\top\Phi \in \mathbb{R}^{H\times H}$. The OLS results assume $\mathrm{rank}(\Phi) = H$, so $n \geq H$ and $G$ is positive definite. Ridge fitting uses $\lambda > 0$ and also applies when this rank condition fails. All actions share the same cases, representation, fitting objective, and regularization strength. Rank deficiency can arise from more feature coordinates than training cases, inactive units, or linearly dependent feature columns. The decomposition algebra also applies to other fixed feature maps, while the experiments examine learned neural representations.

We use $\widehat{Q}$ and $\alpha$ for OLS in Section 3 and Appendix A.2, and write $\widehat{Q}_\lambda$ and $\alpha_\lambda$ when specifying ridge fitting. In Sections 4–5 and Appendices A.4–A.5, $\widehat{Q}$ denotes the chosen fitted score and $\alpha$ its case coefficients. The identities decompose this fitted score; Eq. (9) connects its selected action to the network. The experiments retain the network score-layer bias in $S$ and fit the return models without an additional intercept.

### A.2 Case decomposition and the minimum-norm coefficient vector (Proposition 1)

Using the same learned representation for all actions gives one action-independent case coefficient vector. For action $a$, differentiate the squared fitting loss:

$$\nabla_v \|\Phi v - \mathbf{r}^a\|_2^2 = 2\Phi^\top(\Phi v - \mathbf{r}^a). \tag{A1}$$

The normal equation is $Gv_a = \Phi^\top \mathbf{r}^a$. Full column rank makes $G$ invertible, so substitution into the fitted score gives $\widehat{Q}(x,a) = \phi(x)^\top G^{-1}\Phi^\top \mathbf{r}^a = \alpha(x)^\top \mathbf{r}^a$, where $\alpha(x) = \Phi G^{-1}\phi(x)$. Its $i$th entry

is the coefficient in Eq. (3), since $G^{-1}$ is symmetric. Neither the action index nor the return vector appears in $\alpha(x)$. This identity therefore establishes Eq. (4) simultaneously for every action.

To establish the characterization in Section 3.2, consider

$$\min_{u \in \mathbb{R}^n} \|u\|_2^2 \quad \text{subject to} \quad \Phi^\top u = \phi(x). \tag{A2}$$

The proposed vector is feasible because $\Phi^\top \alpha(x) = GG^{-1}\phi(x) = \phi(x)$. Every other feasible vector has the form $u = \alpha(x) + z$ with $\Phi^\top z = 0$. Since $\alpha(x)$ belongs to the column space of $\Phi$, $\alpha(x)^\top z = \phi(x)^\top G^{-1}\Phi^\top z = 0$. Expanding the squared norm therefore gives

$$\|u\|_2^2 = \|\alpha(x)\|_2^2 + \|z\|_2^2 . \tag{A3}$$

Hence $\|u\|_2^2 \geq \|\alpha(x)\|_2^2$, with equality only for $z = 0$. This proves that $\alpha(x)$ is the unique minimum-norm vector satisfying the representation constraint. The uniqueness concerns this specified representation and OLS objective. It does not require the training returns to determine a unique expansion by themselves.

For any two actions, subtracting their score decompositions yields $\Delta\widehat{Q}(x; a, b) = \sum_{i=1}^n \alpha_i(x)\Delta r_i(a, b) = \sum_{i=1}^n \chi_i(x; a, b)$. This proves the fitted-margin decomposition in Eq. (5). All training cases enter the identity. A displayed list of supporting and opposing cases is a partial view of that sum.

### A.3 Ridge fitting and checkpoint-specific case maps

The experiments use ridge fitting with $\lambda > 0$. For each action, $v_{\lambda,a}$ minimizes $\|\Phi v - \mathbf{r}^a\|_2^2 + \lambda \|v\|_2^2$. The Hessian is $2(G + \lambda I_H)$, and $z^\top (G + \lambda I_H) z = \|\Phi z\|_2^2 + \lambda \|z\|_2^2 > 0$ for every nonzero $z \in \mathbb{R}^H$. Thus the objective has a unique minimizer, including when $\Phi$ is rank deficient. Its normal equation is $(G + \lambda I_H) v_{\lambda,a} = \Phi^\top \mathbf{r}^a$, giving

$$v_{\lambda,a} = (G + \lambda I_H)^{-1} \Phi^\top \mathbf{r}^a. \tag{A4}$$

$$\alpha_\lambda(x) = \Phi (G + \lambda I_H)^{-1} \phi(x). \tag{A5}$$

The fitted ridge score is $\widehat{Q}_\lambda(x, a) = v_{\lambda,a}^\top \phi(x) = \alpha_\lambda(x)^\top \mathbf{r}^a$. Its pairwise margin therefore has contributions $\chi_{\lambda,i}(x; a, b) = \alpha_{\lambda,i}(x)\Delta r_i(a, b)$. Both expansions are exact for this fitted model. The coefficients satisfy

$$\Phi^\top \alpha_\lambda(x) = \phi(x) - \lambda (G + \lambda I_H)^{-1} \phi(x). \tag{A6}$$

Equation (A6) follows from $\Phi^\top \alpha_\lambda = G(G + \lambda I_H)^{-1}\phi$ and $G = (G + \lambda I_H) - \lambda I_H$. The residual distinguishes ridge fitting from the equality constraint in Eq. (A2). When $G$ is invertible, $\alpha_\lambda(x)$ converges to the OLS vector $\alpha(x)$ as $\lambda \to 0$.

The same construction applies at any saved checkpoint $k$. Holding that checkpoint's representation fixed gives

$$\alpha_{k,\lambda}(x) = \Phi_k (G_k + \lambda I_H)^{-1} \phi_k(x), \tag{A7}$$

Here $G_k = \Phi_k^\top \Phi_k$, and $\widehat{Q}_{k,\lambda}(x, a) = \alpha_{k,\lambda}(x)^\top \mathbf{r}^a$. For OLS, set $\lambda = 0$ under full column rank. Each checkpoint receives newly fitted return-model weights. The resulting changes in the case coefficients are examined in Appendix C.2.

### A.4 Conditions for Case-based interpretation and the Nonnegative Approximation

For either fitted model, nonnegative coefficients $\alpha_i(x) \geq 0$ can serve as CBDT relevance weights by setting $\omega_i(x) = \alpha_i(x)$. This choice reproduces every fitted score and pairwise margin. The formulation uses nonnegative weights without a unit-sum constraint; multiplying all weights by a common positive number preserves the selected action. A negative entry prevents that coefficient vector from serving directly as nonnegative relevance weights, while leaving its signed decomposition intact.

Clipping replaces negative coefficients by zero. Define $\alpha_i^+(x) = \max\{\alpha_i(x), 0\}$ and $\alpha_i^-(x) = \max\{-\alpha_i(x), 0\}$, so $\alpha_i(x) = \alpha_i^+(x) - \alpha_i^-(x)$. Let $\nu(x) = \sum_{i=1}^n \alpha_i^-(x)$ and $Q^+(x,a) = \sum_{i=1}^n \alpha_i^+(x) r_i(a)$. If $|r_i(a)| \le R_a$ for every case, then

$$|\widehat{Q}(x,a) - Q^+(x,a)| = \left| \sum_{i=1}^n \alpha_i^-(x) r_i(a) \right| \le \nu(x) R_a. \tag{A8}$$

Cancelling the positive parts gives $\widehat{Q} - Q^+ = -\sum_i \alpha_i^- r_i(a)$. Taking absolute values removes the leading minus sign, and the triangle inequality gives $|\sum_i \alpha_i^- r_i(a)| \le \sum_i \alpha_i^- |r_i(a)| \le \nu R_a$. In the ridge experiments, the negative mass is $\nu_\lambda(x) = \sum_{i=1}^n \max\{-\alpha_{\lambda,i}(x), 0\}$.

For the pairwise clipped margin $\Delta Q^+(x; a, b) = Q^+(x,a) - Q^+(x,b)$, the same argument gives

$$|\Delta\widehat{Q}(x; a, b) - \Delta Q^+(x; a, b)| \le \nu(x) \max_i |\Delta r_i(a, b)|. \tag{A9}$$

Since $\max_i |\Delta r_i(a,b)| \le R_a + R_b$, the action-wise return bounds also control this difference. A small bound supports a close nonnegative approximation on the chosen return scale. A large bound leaves that approximation unresolved; it is not a measurement of the actual clipping error. The observed scale on the bidding task is reported in Appendix D.

If the sum of the clipped coefficients is positive, dividing all of them by that sum preserves the action selected by $Q^+$. This normalization rescales its scores, so Eq. (A8) refers to direct clipping before normalization. The same distinction applies to ridge coefficients.

### A.5 Sufficient condition for decision consistency (Theorem 1)

Fix either fitted model and let $\widehat{a} = a_{\widehat{Q}}(x)$ be its selected action at the query under consideration. Its winning margin is

$$m_{\widehat{Q}}(x) = \widehat{Q}(x, \widehat{a}) - \max_{b \in \mathcal{A},\ b \ne \widehat{a}} \widehat{Q}(x, b). \tag{A10}$$

Set $\varepsilon_{\text{fid}}(x) = \max_{a \in \mathcal{A}} |S(x,a) - \widehat{Q}(x,a)|$. Writing $\delta_a = S(x,a) - \widehat{Q}(x,a)$ gives $|\delta_a| \le \varepsilon_{\text{fid}}(x)$ for every action. For any competitor $b \ne \widehat{a}$,

$$\begin{aligned} S(x,\widehat{a}) - S(x,b) &= \widehat{Q}(x,\widehat{a}) - \widehat{Q}(x,b) + \delta_{\widehat{a}} - \delta_b \\ &\ge \widehat{Q}(x,\widehat{a}) - \widehat{Q}(x,b) - |\delta_{\widehat{a}}| - |\delta_b| \\ &\ge \widehat{Q}(x,\widehat{a}) - \widehat{Q}(x,b) - 2\varepsilon_{\text{fid}}(x). \end{aligned} \tag{A11}$$

The right-hand side is at least $m_{\widehat{Q}}(x) - 2\varepsilon_{\text{fid}}(x)$. Under Eq. (9), it is strictly positive for every competitor. Hence $a_S(x) = \widehat{a} = a_{\widehat{Q}}(x)$. The proof uses the fitted scores and their error, so it applies to both OLS and ridge.

The factor two accounts for errors of opposite signs on the selected action and its competitor. Decision consistency establishes the same maximizing action. It leaves the ordering among the remaining actions unrestricted, and it does not equate the numerical margins of the two models. The strict condition can fail even when their selected actions agree. Coverage of the sufficient condition and observed agreement outside that set are evaluated in Appendix E.

This result is independent of coefficient nonnegativity. A signed case decomposition can have a decision with guaranteed agreement. A CBDT interpretation of that decision requires both the nonnegativity condition in Section 4 and decision consistency.

**Tightness.** Let $\varepsilon > 0$ and consider two actions with fitted margin $m > 0$. Set $S(x,a) - \widehat{Q}(x,a) = -\varepsilon$ and $S(x,b) - \widehat{Q}(x,b) = \varepsilon$. Then $\varepsilon_{\text{fid}}(x) = \varepsilon$ and the network margin is $m - 2\varepsilon$. At $m = 2\varepsilon$ the network is tied. For any $c < 2$, choosing $\max\{c, 0\}\varepsilon < m < 2\varepsilon$ satisfies $m > c\varepsilon$ but strictly reverses the decision. Hence no smaller constant provides the same uniform guarantee from the score-error bound alone.

### A.6 Coordinate changes and score scale

For OLS under full column rank, let $B \in \mathbb{R}^{H\times H}$ be invertible and define $\phi_B(x) = B\phi(x)$. Since training representations are rows, $\Phi_B = \Phi B^\top$ and $G_B = BGB^\top$. Therefore

$$\Phi_B G_B^{-1}\phi_B(x) = \Phi B^\top B^{-\top} G^{-1} B^{-1} B\phi(x) = \alpha(x). \tag{A12}$$

Thus the OLS coefficients, their signs, and their case-margin contributions are unchanged. This proves Corollary 1. The kernel is also unchanged because $(B\phi(x))^\top (BGB^\top)^{-1} B\phi(x') = K_G(x, x')$. If the uncentered second moment is whitened so that $G = nI_H$, then $\alpha_i(x) = \phi(x)^\top \phi(x_i)/n$. This inner product can be negative; whitening does not imply nonnegative relevance.

With ridge fitting, an invertible coordinate change can alter the coefficients when the penalty remains $\lambda I_H$. For orthogonal $B$, $G_B + \lambda I_H = B(G + \lambda I_H)B^\top$ because $BB^\top = I_H$. Substitution into Eq. (A5) then gives the same ridge coefficients.

Decision consistency is also invariant to a common positive score scale. Replacing both score systems by $S'(x, a) = cS(x, a) + d(x)$ and $\widehat{Q}'(x, a) = c\widehat{Q}(x, a) + d(x)$, with $c > 0$ and an action-independent offset $d(x)$, multiplies both $m_{\widehat{Q}}$ and $\varepsilon_{\text{fid}}$ by $c$. Equation (9) is unchanged.

### A.7 Leverage and the sum of case coefficients

At a training situation, its own case coefficient is $\alpha_i(x_i) = [P_\Phi]_{ii}$, where $P_\Phi = \Phi G^{-1}\Phi^\top$ is the OLS projection matrix. Thus these self-weights equal the statistical leverages. Since $\operatorname{tr}(P_\Phi) = \operatorname{tr}(G^{-1}\Phi^\top\Phi) = H$, they sum to $H$. For ridge fitting, $P_{\Phi,\lambda} = \Phi(G + \lambda I_H)^{-1}\Phi^\top$ gives $\sum_i \alpha_{\lambda,i}(x_i) = \operatorname{tr}(G(G + \lambda I_H)^{-1})$. This trace is the effective dimension of the fitted linear model. A large leverage indicates sensitivity of a fitted training value to its own return, while support for a particular action also depends on the return contrast.

If the representation includes a constant coordinate equal to one at every training and query situation and retains full column rank, the OLS constraint $\Phi^\top\alpha(x) = \phi(x)$ implies $\sum_i \alpha_i(x) = 1$. Nonnegative coefficients would then form a convex average of the case returns. The unit-sum identity alone does not ensure nonnegativity. This observation concerns an augmented representation and does not change the experimental fits.

## B Experimental Setting and Evaluation

The experiments evaluate three properties of the recovered case structure: correspondence with known case contributions, case-based interpretation, and consistency with the original network decision. The controlled setting supplies a known support function. Adult, Default Credit, and the bidding task use real-world decision inputs and constructed action returns. Each case therefore has the complete return vector required by the analysis.

### B.1 Tasks and fitted models

The controlled environment contains 20 latent CBDT cases in two dimensions, three actions, and Gaussian similarity of bandwidth 1. The latent case locations and action-return parameters are sampled once per seed. Their weighted support function generates returns for 500 training inputs and 200 test inputs. The 500 sampled training problems form the decomposition's case collection; the 20 latent cases define the environment. This setting tests agreement with a known support function and a specified margin reference; the two case collections are distinct.

Adult uses six numeric covariates from 30,162 complete records of the University of California, Irvine (UCI) dataset (Becker & Kohavi, 1996). Default uses 23 coded covariates from 30,000 records in Default of Credit Card Clients (Yeh, 2009; Yeh & Lien, 2009). Both use seed-specific random 80/20 splits. The bidding task uses 2017 PJM load data (PJM Interconnection, 2017). Before task construction, finer-interval observations were aggregated into half-hour intervals, yielding 365 ordered days with 48 slots per day. Current load, one-hour-lagged load, and cyclic time features form each input. Dropping the first two slots of each day gives 16,790 instances, split chronologically 80/20. Network configurations are listed in Table B1.

Table B1: Network configurations and data splits for the controlled setting and real-data tasks. All networks have three action outputs. The wider PJM network is used for coefficient-sign analysis and the individual decision example; the narrower network is used for cross-task consistency. Train/test counts refer to the task inputs after preprocessing.

| Experiment / task | Hidden widths | Max. iterations | Train / test |
|---|---|---|---|
| Controlled CBDT | 32, 16 | 2,000 | 500 / 200 |
| PJM consistency | 32, 16 | 2,000 | 13,432 / 3,358 |
| PJM signs and case example | 128, 64, 32 | 2,000 | 13,432 / 3,358 |
| Adult | 64, 32, 16 | 2,000 | 24,129 / 6,033 |
| Default | 64, 32, 16 | 800 | 24,000 / 6,000 |

Every network is a multilayer perceptron (MLP) with rectified linear unit (ReLU) hidden layers and three linear outputs and is trained on squared return error with adaptive moment estimation (Adam). The initial learning rate is $10^{-3}$ and weight regularization is $10^{-4}$. Input standardization uses the training partition. Default selects a stopping point using a 10% validation subset and patience 20; the other final-model runs use training-loss stopping criteria within their stated iteration limits.

After training, the final hidden representation is fixed. Each action receives a ridge return model with $\lambda = 10^{-5}$ and no additional intercept. All actions share the same representation, case collection, and regularization, so their recovered case coefficients can be compared through the action-specific return contrasts.

### B.2 What the constructed returns represent

Adult's actions represent conservative, standard, and expanded approval. Their base returns depend on income, education, and capital gains, with fixed Gaussian perturbations of standard deviation 0.05 applied to all action-return entries. Default's reduce, maintain, and increase actions use a constructed risk score combining the observed default label, repayment delays, credit limit, and repayment-to-bill ratio. The Standard base return decreases with absolute standardized risk, giving it a distinct preference region. The returns include fixed Gaussian perturbations of standard deviation 0.04. The maintain action is labeled Standard in the reported figures and tables. The normalization used to define Default's utility is computed before splitting; network-input normalization is fitted on the training partition.

The bidding task attaches returns to low, medium, and high bids at 25, 45, and 70. A load-dependent synthetic clearing price determines whether a bid is accepted and its resulting return. The real data supply the covariates in these tasks; the action utilities are defined by the experiment. The stored return tables and construction code specify these utilities.

The explicit CBDT rule in Figure 3 provides a reference case-based decision process by combining the returns of the 50 most similar training cases with nonnegative Gaussian input-based weights. This comparison evaluates whether neural network decisions and case-based decisions can exhibit similar outcomes while relying on different case contribution patterns. The decision-consistency experiments compare the recovered case-based formulation $\widehat{Q}_\lambda$ with the original network scores $S$.

### B.3 Reference contributions and distinct evaluation quantities

The controlled margin reference uses the generator-selected action $a_{\mathrm{gt}}$ and runner-up $b_{\mathrm{gt}}$. For each of the 500 sampled training cases, it is

$$\chi_i^{\mathrm{gt}}(x) = \exp(-\|x - x_i\|_2^2/2)\,[r_i(a_{\mathrm{gt}}) - r_i(b_{\mathrm{gt}})]. \tag{B1}$$

Every compared method uses this action pair. Equation (B1) defines a Gaussian relevance-and-return reference over the 500 sampled training cases. The generating support function instead uses 20 latent cases. The comparison measures alignment with the specified reference over training cases.

Pearson correlation compares signed scores across all training cases. Top-30 Jaccard compares the case sets with the largest absolute method and reference scores. Normalized discounted cumulative

gain (nDCG) uses absolute reference contributions as gains. The signed Top-30 measure is direction agreement at the method's 30 retrieved indices; it does not also require those indices to belong to the reference Top-30 set.

Two additional case statistics answer different questions. In the representation ablation, Top-30 action consistency uses the $|\alpha_\lambda|$-weighted majority action of the 30 largest absolute coefficients and compares it with the generator's selected action. In the checkpoint analysis, Jaccard compares the Top-30 absolute-coefficient sets at the current and final checkpoints.

Reconstruction $R^2$ measures how closely the fitted scores reproduce the network scores on the evaluation inputs. Squared errors are pooled across inputs and actions, and each network action is centered separately:

$$R^2 = 1 - \frac{\sum_{j=1}^{N} \sum_{a \in \mathcal{A}} (S(x_j, a) - \widehat{Q}(x_j, a))^2}{\sum_{j=1}^{N} \sum_{a \in \mathcal{A}} (S(x_j, a) - \bar{S}_a)^2} \tag{B2}$$

Here, $N$ is the number of evaluation inputs and $\bar{S}_a = \frac{1}{N} \sum_{j=1}^{N} S(x_j, a)$. The implementation adds $10^{-12}$ to the denominator for numerical stability. Support correlation compares the fitted scores with the controlled support function. Consistency coverage is the fraction satisfying Eq. (9). The real-task score correlations in Appendix E compare fitted scores with network scores.

### B.4 Aggregation and uncertainty

Five training seeds, 42–46, are the independent units. The margin comparison uses 96 test inputs per seed, sampled before scoring. Representation and checkpoint tables report means $\pm$ sample standard deviations (SD). Bracketed intervals are Student-$t$ 95% confidence intervals (CI) over five seed summaries; paired comparisons use the five within-seed differences. Noise replicates are averaged within seeds before inference.

The anonymous supplementary code attachment contains the experiment scripts and their comma-separated values (CSV) and JavaScript Object Notation (JSON) summaries. Its recorded environment uses Python 3.13.9, NumPy 2.3.5, and scikit-learn 1.7.2. The appendix reports the controls that determine the interpretation of each result; implementation parameters and individual run outputs remain in that package.

## C Learned Representations and Case Decomposition

### C.1 Representation ablation

The ablation tests which representation preserves the information required to recover the case structure of fitted decisions. The training cases and action returns remain fixed within each seed. We compare standardized raw inputs, a random ReLU feature map of width 16, an untrained 32,16 ReLU network, and the final hidden layer of the trained 32,16 network. All four receive the same ridge return-model fit.

Random features use Gaussian weights scaled by the square root of the input dimension and Gaussian biases with standard deviation 0.1. The untrained network uses Xavier-uniform weights and zero biases. Reconstruction and consistency coverage for every representation are evaluated against the same trained network scores. This common reference makes their fidelity values comparable.

Learning improves the correspondence between the fitted scores and the trained network (Table C1). The same ridge objective admits an algebraic case decomposition for every representation, so the reconstruction contrast concerns the information retained in the features. Random features capture part of the smooth support function, but the learned representation more closely preserves the score organization needed for decision consistency.

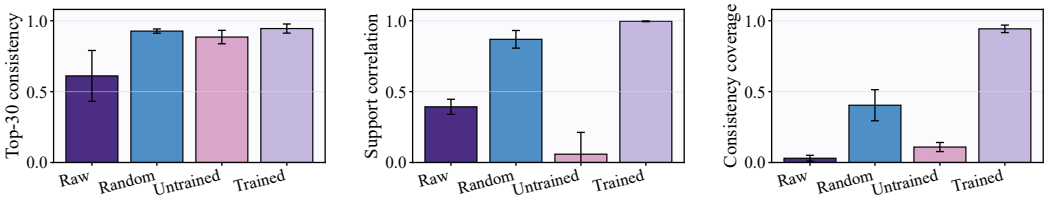
Retrieving cases with matching preferred actions does not require accurate recovery of their aggregate support. This distinction explains why case-action consistency can remain high for a representation with weak support reconstruction (Table C2; Figure C1). The coefficient signs impose a separate condition: the recovered structure in this setting is a signed case decomposition, even when its decisions and support scores closely match the reference.

Table C1: Reconstruction and decision consistency across representations. Mean ± SD over five seeds. The trained representation achieves near-exact network-score reconstruction ($R^2 = 0.9998 \pm 0.0001$), high support recovery ($0.996 \pm 0.003$), and the highest coverage ($0.943 \pm 0.026$), while random features recover part of the support structure with lower coverage ($0.404 \pm 0.110$). The unrounded mean support correlation for the trained representation is 0.9962.

| **Representation** | **Network $R^2$** | **Support r** | **Coverage** |
|---|---|---|---|
| Raw covariates | $-1.0648 \pm 0.3527$ | $0.393 \pm 0.053$ | $0.030 \pm 0.021$ |
| Random features | $0.7663 \pm 0.1187$ | $0.869 \pm 0.062$ | $0.404 \pm 0.110$ |
| Untrained network | $-0.6710 \pm 0.2541$ | $0.059 \pm 0.154$ | $0.109 \pm 0.032$ |
| Trained network | $0.9998 \pm 0.0001$ | $0.996 \pm 0.003$ | $0.943 \pm 0.026$ |

Table C2: Case-action consistency and coefficient sign across representations. Mean ± SD over five seeds. The untrained network attains Top-30 action consistency of 0.885 despite support correlation of 0.059 in Table C1. No evaluated query has a fully nonnegative coefficient vector, including queries evaluated with the trained representation.

| **Representation** | **Top-30 action consistency** | **Negative mass** | **Fully nonnegative queries** |
|---|---|---|---|
| Raw covariates | $0.611 \pm 0.179$ | $0.506 \pm 0.011$ | $0.000 \pm 0.000$ |
| Random features | $0.927 \pm 0.015$ | $0.713 \pm 0.086$ | $0.000 \pm 0.000$ |
| Untrained network | $0.885 \pm 0.047$ | $0.513 \pm 0.018$ | $0.000 \pm 0.000$ |
| Trained network | $0.945 \pm 0.032$ | $0.746 \pm 0.034$ | $0.000 \pm 0.000$ |




(a) Action consistency (b) Support correlation (c) Consistency coverage

Figure C1: Representation ablation on the controlled CBDT task. Panels report (a) weighted Top-30 action consistency, (b) support correlation, and (c) decision-consistency coverage. The untrained network retains high action consistency but weak support correlation; the trained representation has the highest means across all three measures. Error bars show SD across five seeds. Tables C1 and C2 give the numerical results.

### C.2 Changes across training checkpoints

The trajectory uses checkpoints at epochs 1, 5, 10, 25, 50, 100, 250, 500, 1000, and 2000, refitting the return models with each fixed representation. Table C3 reports selected checkpoints. The final-model analysis requires only the final network state. The overlap of the largest-coefficient case sets measures how the situation-to-case mapping develops relative to the final fit. Comparing this overlap with score reconstruction tests whether faithful predictions and a stable case map emerge at the same stage of learning (Table C3).

Table C3: Training-checkpoint trajectory from epoch 1 to epoch 2000. Train MSE is a seed mean; other entries are mean ± SD over five seeds. Reconstruction reaches $R^2 = 0.981$ at epoch 500 and approximately 1 at epoch 2000. The Top-30 absolute-coefficient set has mean Jaccard overlap of 0.539 and 0.708 with the final set at epochs 500 and 1000, respectively. Its final overlap of 1 is a self-comparison.

| Epoch | Train MSE | Network $R^2$ | Action agreement | Top-30 coefficient Jaccard |
|---|---|---|---|---|
| 1 | 184.748 | $-2668.125 \pm 878.130$ | $0.109 \pm 0.070$ | $0.427 \pm 0.033$ |
| 25 | 169.058 | $-2368.707 \pm 3729.723$ | $0.528 \pm 0.246$ | $0.389 \pm 0.066$ |
| 100 | 84.705 | $-7.864 \pm 8.546$ | $0.780 \pm 0.070$ | $0.434 \pm 0.013$ |
| 500 | 2.748 | $0.981 \pm 0.009$ | $0.935 \pm 0.024$ | $0.539 \pm 0.023$ |
| 1000 | 0.335 | $0.999 \pm 0.000$ | $0.985 \pm 0.015$ | $0.708 \pm 0.043$ |
| 2000 | 0.130 | $1.000 \pm 0.000$ | $0.992 \pm 0.004$ | $1.000 \pm 0.000$ |

Early reconstruction errors are large relative to the network's small score variance. As training progresses, accurate score reconstruction precedes stabilization of the cases receiving the largest coefficients (Figure C2). A decision score aggregates many contributions, so similar aggregate scores can be supported by different leading case sets. The trajectory therefore separates fidelity of the recovered scores from stability of the recovered situation-to-case mapping.

The checkpoint run retains weights between successive training calls and restarts optimizer state. It is separate from the final-model ablation.

## D Conditions for Case-Based Interpretation

### D.1 Comparison with an explicit CBDT Reference

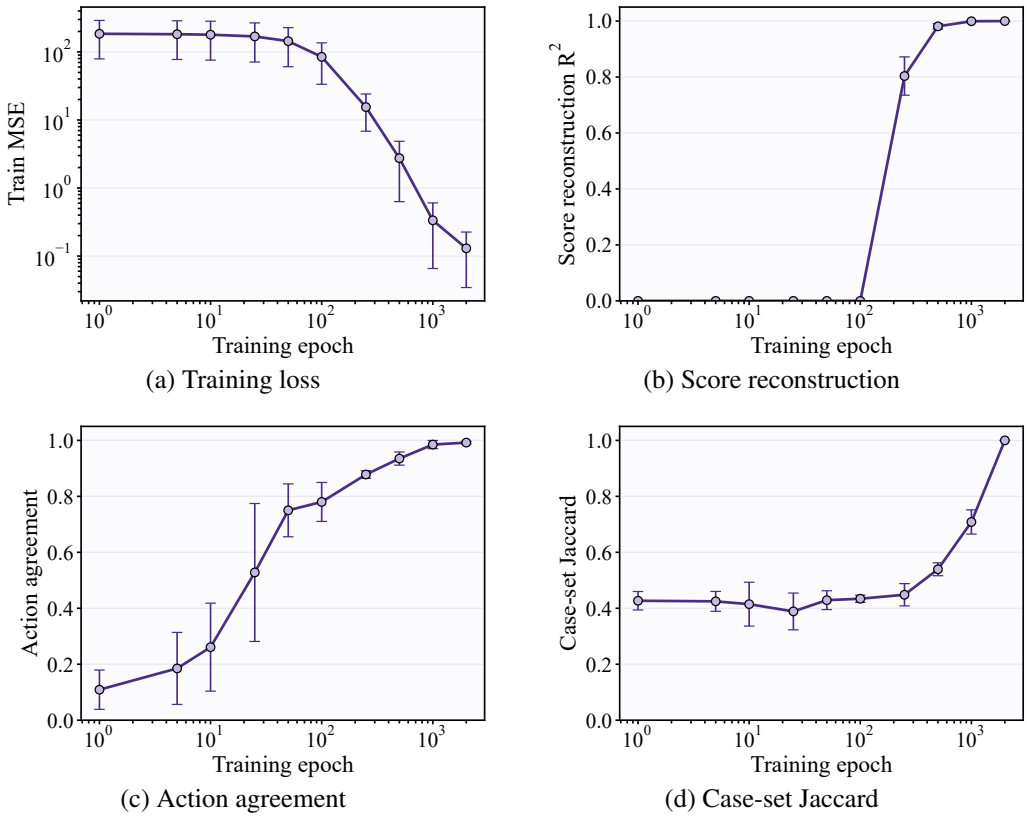
Figure 3 compares the neural network with a separate nonnegative CBDT rule on the bidding task. This comparison concerns $U$ and $S$, whereas the fitted-model comparison in Appendix E concerns $\widehat{Q}_\lambda$ and $S$. Distinguishing these pairs matters because agreement with an explicit CBDT rule does not identify the case structure induced by the network's representation.

The low Aggressive correlation shows that frequent agreement on the winning action can coexist with weak agreement on one action's score. It does not establish the signs of the representation-induced coefficients. We examine those coefficients directly to evaluate the condition in Section 4.

### D.2 Negative coefficients and the scale of clipping

To examine whether the recovered case structure satisfies the case-based interpretation condition, we evaluate 50 randomly sampled test inputs per seed using the wider bidding network in Table B1. For each input, $\nu_\lambda(x) = \sum_{i=1}^{n} \max\{-\alpha_{\lambda,i}(x), 0\}$ is its negative mass. The negative-coefficient rate is the proportion of negative entries, averaged over queries. Both quantities are summarized in Table D1.

The sign pattern in Table D1 prevents a direct nonnegative-relevance interpretation under the condition in Section 4. The fitted scores still admit the signed decomposition derived from the ridge objective. Nonnegativity determines whether the recovered coefficients admit a CBDT interpretation. The algebraic decomposition holds for coefficients of either sign.




(a) Training loss (b) Score reconstruction

(c) Action agreement (d) Case-set Jaccard

Figure C2: Case maps across training checkpoints. (a) Training loss, (b) network-score reconstruction, (c) fitted–network action agreement, and (d) overlap with the final Top-30 absolute-coefficient set. Reconstruction and action agreement improve while the leading case set continues to change. Negative $R^2$ values are displayed at zero in panel (b); Table C3 reports their original values. Error bars show SD across five seeds; panel (a) plots the seed-mean MSE.

Table D1: Coefficient signs on 50 bidding queries per seed. Mean $\pm$ SD across five seeds. All 250 inputs have at least one negative coefficient. The action-averaged clipping bound $E_\lambda^{\text{avg}}$ defined in the text has pooled mean 200.4 and median 184.1; its median ratios to the fitted score range and winning margin are 39.0 and 54.0, respectively.

| Quantity | **Value** |
|---|---|
| Negative-coefficient rate | $0.440 \pm 0.007$ |
| Mean negative mass | $0.877 \pm 0.047$ |
| Mean within-seed median negative mass | $0.840 \pm 0.050$ |
| Mean within-seed 95th percentile negative mass | $1.313 \pm 0.171$ |
| Fraction of fully nonnegative queries | $0.000 \pm 0.000$ |

To compare the clipping bound with the fitted score scale, set $R_a = \max_i |r_i(a)|$ and $E_\lambda^{\text{avg}}(x) = \nu_\lambda(x) \sum_{a \in \mathcal{A}} R_a / |\mathcal{A}|$. This is the action-averaged upper bound in Eq. (A8), evaluated with the ridge coefficients. Its scale relative to the fitted scores leaves the actual clipping error unresolved. The individual decision below therefore retains the signed coefficients.

### D.3 Supporting and opposing cases for one decision

The following example illustrates how the recovered case structure traces an individual decision back to supporting and opposing cases. For the query in Table D2, let $a_0, a_1, a_2$ denote the low, medium, and high bids, respectively. In this order, the network and fitted score vectors are

$$S(x, \cdot) = [13.908,\ 7.353,\ -2.357], \tag{D1}$$

$$\widehat{Q}_\lambda(x, \cdot) = [14.062,\ 6.068,\ -2.596]. \tag{D2}$$

The fitted winning margin exceeds twice the maximum score discrepancy, so Eq. (9) guarantees agreement on the selected action. Table D2 resolves that margin into supporting and opposing contributions for $a_0$ against the runner-up $a_1$.

Table D2: Three supporting and three opposing cases for $a_0$ against runner-up $a_1$, using seed 46 and test input 2325. The unrounded fitted margin is 7.9938 and the maximum score discrepancy is 1.2852. All six displayed coefficients are positive; return contrasts distinguish support from opposition. For case 1221, $0.005749 \times (-70.152) \approx -0.4033$. The displayed contributions sum to $-0.7045$; the remaining cases contribute 8.6983, completing the fitted margin.

| Role | Case | Coefficient | Return contrast | Contribution |
|---|---|---|---|---|
| Support | 1773 | +0.005908 | +19.894 | +0.1175 |
| Support | 1452 | +0.005683 | +19.843 | +0.1128 |
| Support | 1312 | +0.005898 | +19.056 | +0.1124 |
| Oppose | 1221 | +0.005749 | -70.152 | -0.4033 |
| Oppose | 3890 | +0.005544 | -63.990 | -0.3547 |
| Oppose | 8396 | +0.005302 | -54.534 | -0.2892 |

A case supports the selected action through the product of its coefficient and return contrast. Positive coefficients can therefore contribute on either side of the comparison when cases favor different actions (Table D2). Reading the coefficient sign alone would lose this action-dependent meaning.

The balance of the displayed cases also differs from the balance of the full collection. A short list exposes strong contributions in each direction, but the decision follows from their sum over all training cases. The margin accounting in Table D2 links that local explanation to the choice satisfying the consistency condition.

The example was selected from inputs satisfying the condition with positive-coefficient cases on both sides of the action comparison. The selection requires at least two of the three displayed supporting cases to have positive coefficients and positive return contrasts, and at least one of the three opposing cases to have a positive coefficient and a negative return contrast. This pattern occurs in 8,314 of 15,399 candidate inputs satisfying the condition across the five wider-network runs. Candidate extraction requires three supporting and three opposing cases and retains 3,165, 3,088, 3,017, 3,104, and 3,025 inputs for seeds 42–46, respectively. These counts belong to the case-record experiment, whose architecture differs from the consistency experiment in Appendix E.1.

### D.4 Nonnegative reconstruction and fitted-score deviation

The nonnegative fit in Section 4.1 minimizes $\|\Phi^\top u - \phi(x)\|_2^2$ over $u \geq 0$. The reconstructed representation is a projection onto the cone generated by the training representations. Several coefficient vectors can attain the same minimum, so $\beta(x)$ denotes the nonnegative least-squares solver's returned vector. Its criterion concerns representation reconstruction. It does not minimize distance to $\alpha(x)$ or deviation in decision scores.

To see why representation and score fidelity can differ, let $R = [\mathbf{r}^a]_{a\in\mathcal{A}}$, $V_\lambda = (G + \lambda I_H)^{-1}\Phi^\top R$, and $E_\lambda = R - \Phi V_\lambda$ be the training-return residual matrix. Write $d_\beta(x) = \Phi^\top\beta(x) - \phi(x)$ and $Q_\beta(x,a) = \beta(x)^\top\mathbf{r}^a$. Direct substitution gives

$$(Q_\beta(x,a) - \widehat{Q}_\lambda(x,a))_{a\in\mathcal{A}} = V_\lambda^\top d_\beta(x) + E_\lambda^\top\beta(x). \tag{D3}$$

Even with exact representation reconstruction, the residual term can change the scores. Under full-rank OLS, the same identity holds with $\lambda = 0$. This explains the score deviations in Table D3 without attributing them solely to representation error.

We use the three consistency-task configurations in Table B1, including the 32–16 PJM network, and seeds 42–46. Each seed uses 200 held-out queries per task, sampled without replacement. The signed reference coefficients use $\lambda = 10^{-5}$. All sampled coefficient vectors contain negative entries. Reported quantities are the negative-mass fraction $\nu_\lambda/\|\alpha_\lambda\|_1$, relative reconstruction residual $\|d_\beta(x)\|_2/\|\phi(x)\|_2$, and relative score deviation $\max_a|Q_\beta(x,a) - \widehat{Q}_\lambda(x,a)|/\max_a|\widehat{Q}_\lambda(x,a)|$. Computations add $10^{-12}$ to denominators. Queries are averaged within seeds before computing means and Student-$t$ intervals.

Table D3: Nonnegative representation reconstruction over five seeds. Every sampled coefficient vector contains negative entries. Small representation residuals coexist with substantially larger score deviations. Brackets report 95% seed-level intervals.

| Task | Negative fraction | Reconstruction residual | Score deviation |
|---|---|---|---|
| PJM | 0.282 | 0.0006 [0.0004, 0.0007] | 0.420 [0.396, 0.444] |
| Adult | 0.255 | 0.0019 [0.0012, 0.0025] | 0.203 [0.182, 0.224] |
| Default | 0.250 | 0.0013 [0.0003, 0.0024] | 0.374 [0.346, 0.403] |

For comparison, partial clipping uses $\alpha_\lambda^{(t)} = \alpha_\lambda + t\alpha_\lambda^-$ for $t \in \{0, 0.25, 0.5, 0.75, 1\}$, where $\alpha_\lambda^-$ is the magnitude of its negative part. Its score deviation is exactly linear in $t$ for each query, since the score change equals $t\sum_i \alpha_{\lambda,i}^- r_i(a)$. Complete clipping gives mean relative deviations of 0.631, 0.581, and 0.351 for PJM, Adult, and Default. The nonnegative reconstruction fit yields 0.420, 0.203, and 0.374, respectively. Its reconstruction objective therefore does not uniformly improve score fidelity over clipping.

We also compare selected actions before and after nonnegative reconstruction. On the same 200-query samples, mean disagreement is 14.7% [8.8%, 20.6%] for Adult, 18.9% [14.5%, 23.3%] for Default, and 28.1% [23.5%, 32.7%] for PJM. A separate 300-query sample per seed evaluates whether those changes improve decisions. Accuracy is measured against the action maximizing the constructed return vector, and the reference is the signed ridge fit. Table D4 shows lower mean accuracy after nonnegative reconstruction in every task. These results measure the cost of the specified reconstruction procedure. Other nonnegative coefficient choices can induce different scores. The relative accuracy decreases reported in the main text divide the difference between the five-seed mean accuracies by the signed fit's mean accuracy.

Table D4: Decision accuracy before and after nonnegative reconstruction on 300 queries per task and seed. Loss is signed minus nonnegative accuracy. Brackets give paired 95% intervals over five seeds. All three loss intervals are above zero.

| Task | Signed fit | Nonnegative fit | Accuracy loss |
|---|---|---|---|
| PJM | 0.655 | 0.596 | 0.059 [0.008, 0.111] |
| Adult | 0.798 | 0.765 | 0.033 [0.003, 0.063] |
| Default | 0.730 | 0.684 | 0.046 [0.015, 0.077] |

# E Decision Consistency and Return Error

## E.1 Coverage of the sufficient condition

The decision-consistency experiment evaluates Eq. (9) on all held-out inputs of each task (Table E1). For PJM, seeds 42–46 each have 3,358 held-out inputs, with 3,228, 3,219, 3,224, 3,198, and 3,164 inputs satisfying the condition, respectively. The total is 16,033 of 16,790 evaluations, giving coverage 0.954914, rounded to 95.5%. The consistency condition compares the winning margin with twice the score error. Its value is therefore input-specific: an error small enough to preserve one decision may change another when competing actions have nearly equal scores.

Table E1: Decision consistency across tasks. Means and 95% seed-level intervals over five seeds. Every input satisfying the condition has matching fitted and network actions in every task and seed. Coverage ranges from 0.799 to 0.955, while agreement among remaining inputs ranges from 0.787 to 0.826. The seed-mean ratio of disagreement among remaining inputs to overall disagreement is 22.67 for PJM, 12.84 for Adult, and 5.43 for Default.

| Task | Overall agreement | Coverage | Condition not met agreement |
|---|---|---|---|
| PJM | 0.992 [0.987, 0.997] | 0.955 [0.945, 0.965] | 0.826 [0.753, 0.899] |
| Adult | 0.983 [0.976, 0.990] | 0.919 [0.896, 0.942] | 0.791 [0.757, 0.825] |
| Default | 0.953 [0.914, 0.991] | 0.799 [0.721, 0.877] | 0.787 [0.659, 0.915] |

Eq. (9) identifies inputs on which the fitted and network decisions agree. Additional matching decisions occur among remaining inputs, which also contain all observed disagreements.

Let $A_{\mathrm{all}}$ and $A_{\mathrm{unc}}$ be agreement rates over all queries and remaining queries, respectively. The reported disagreement ratio is $(1 - A_{\mathrm{unc}})/(1 - A_{\mathrm{all}})$ within each seed. Since no query satisfying the condition disagrees, this ratio is the reciprocal of the fraction not satisfying the condition whenever disagreements occur. It describes how the sufficient condition partitions these evaluations.

## E.2 Action-wise score reconstruction

Action-wise reconstruction localizes differences that an aggregate agreement rate can hide (Table E2). In particular, a model can preserve many winning actions while approximating one candidate action less closely. Inspecting the full score vector therefore helps explain where decision consistency may become harder to establish.

Table E2: Fitted–network score correlation by action. Means and 95% seed-level intervals over five seeds. Adult's mean correlations range from 0.996 to 0.999. Default's Standard action has a lower mean correlation of 0.956, whereas Reduce and Increase are both approximately 0.998. Across 30,000 held-out Default evaluations, the network selects Reduce, Standard, and Increase 5,435, 15,582, and 8,983 times, respectively. The fitted model selects them 5,582, 14,736, and 9,682 times.

| Task | Action | Pearson r [95% CI] |
|---|---|---|
| Adult | Conservative | 0.997 [0.995, 0.999] |
| Adult | Standard | 0.996 [0.995, 0.997] |
| Adult | Expanded | 0.999 [0.998, 1.000] |
| Default | Reduce | 0.998 [0.997, 0.999] |
| Default | Standard | 0.956 [0.916, 0.996] |
| Default | Increase | 0.998 [0.997, 0.999] |

The weaker Standard-action correspondence accompanies the smaller coverage on Default. Correlation summarizes variation across inputs; the consistency condition uses the maximum absolute score

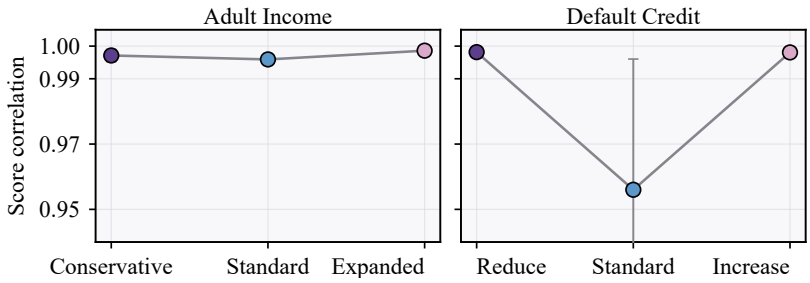



Figure E1: Fitted–network score correlation by action. Adult correlations are nearly identical, while Default varies across actions. Table E2 gives seed-level intervals.

discrepancy at each input. Thus the two measures describe different aspects of fidelity. Figure E1 summarizes the action-wise correlations.

### E.3 Sensitivity to errors in training returns

Estimated returns can change a fitted decomposition even when the representation is held fixed. To isolate this effect on the controlled task, we perturb the training returns by

$$\tilde{r}_i(a) = r_i(a) + \sigma_r \sigma_a \eta_{i,a} \tag{E1}$$

$$\sigma_a = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(r_i(a) - \bar{r}_a)^2}$$

$$\bar{r}_a = \frac{1}{n}\sum_{i=1}^{n} r_i(a)$$

Here, $\tilde{r}_i(a)$ denotes the perturbed return, $\sigma_a$ is the sample standard deviation of the unperturbed training returns for action $a$, and $\sigma_r \in \{0, 0.05, 0.10, 0.20, 0.40\}$ controls the noise magnitude. The variables $\eta_{i,a} \sim \mathcal{N}(0,1)$ are independent. Each nonzero level has ten noise replicates per training seed. The network, representation, data split, and test inputs remain fixed. We refit the return models using the perturbed return table and compare them with the original network scores.

The coefficients $\alpha_\lambda(x)$ remain unchanged because they depend on the representation, not on the returns. Case-margin scores change through the perturbed return contrasts. For their Pearson and Jaccard comparisons, we retain each clean model's selected-versus-runner-up action pair. Replicate-level results are averaged within seeds before forming confidence intervals.

The fitted decision and overall contribution pattern are less sensitive to return perturbations than the identities of the highest-ranked cases (Table E3; Figure E2). With the representation and case coefficients fixed, the contribution changes arise through the return contrasts. Each perturbed fit retains its algebraic decomposition, while consistency with the original network depends on the score discrepancy produced by the perturbation.

### E.4 Decision accuracy under margin-to-error selection

To distinguish fidelity from task performance, we compare the network action with the action maximizing each held-out input's constructed return vector. This reference uses the task-defined returns in Appendix B.2. The consistency condition still tests agreement between the fitted model and the network. The accuracy comparison evaluates an additional empirical relationship.

The five seeds, data splits, fitted models, and held-out inputs match Appendix E.1. Table E4 reports network accuracy overall and within inputs satisfying the condition and the remaining inputs. The ratio of error among remaining inputs to overall error averages 1.46 for PJM, 3.22 for Adult, and 1.73 for Default. These ratios measure task errors. The ratios in Appendix E.1 measure disagreement between the fitted model and the network. All queries are averaged within seeds before forming Student-$t$ intervals.

Table E3: Sensitivity to errors in training returns on the controlled task. Means and 95% seed-level intervals over five seeds, with ten noise replicates averaged within each seed at nonzero levels. At $\sigma_r = 0.20$, action agreement is 0.964 and contribution correlation is 0.953, while Top-30 Jaccard is 0.687. At $\sigma_r = 0.40$, Jaccard decreases to 0.505 and consistency coverage to 0.826. Contribution comparisons retain the clean model's action pair.

| Noise | Action agreement | Coverage | Contribution Pearson | Top-30 Jaccard |
|---|---|---|---|---|
| 0.00 | 0.988 [0.977, 0.999] | 0.943 [0.910, 0.976] | 1.000 [1.000, 1.000] | 1.000 [1.000, 1.000] |
| 0.05 | 0.984 [0.974, 0.994] | 0.934 [0.906, 0.962] | 0.997 [0.995, 0.998] | 0.901 [0.869, 0.933] |
| 0.10 | 0.977 [0.969, 0.985] | 0.918 [0.899, 0.937] | 0.987 [0.981, 0.993] | 0.819 [0.768, 0.871] |
| 0.20 | 0.964 [0.953, 0.976] | 0.882 [0.862, 0.901] | 0.953 [0.932, 0.974] | 0.687 [0.620, 0.754] |
| 0.40 | 0.945 [0.936, 0.953] | 0.826 [0.806, 0.847] | 0.855 [0.803, 0.908] | 0.505 [0.435, 0.575] |

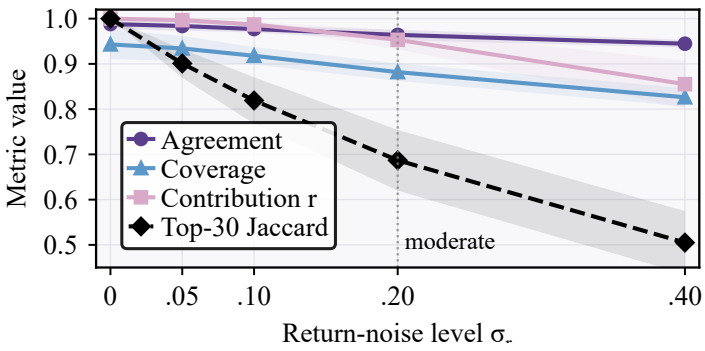



Figure E2: Sensitivity to training-return noise with the network and representation held fixed. Action agreement and contribution correlation decline more slowly than Top-30 case-set overlap. Consistency coverage also decreases as noise increases. The noise scale $\sigma_r$ is relative to each action's training-return SD. Table E3 reports means and 95% seed-level intervals.

Table E4: Network accuracy against task-defined optimal actions. Means and 95% intervals over five seeds. Inputs satisfying the condition have higher accuracy in each task, although the condition alone guarantees fitted–network agreement.

| Task | Overall | Condition met | Condition not met |
|---|---|---|---|
| PJM | 0.671 [0.663, 0.680] | 0.678 [0.670, 0.687] | 0.521 [0.465, 0.577] |
| Adult | 0.798 [0.793, 0.803] | 0.837 [0.825, 0.850] | 0.350 [0.280, 0.420] |
| Default | 0.727 [0.711, 0.743] | 0.776 [0.767, 0.786] | 0.527 [0.463, 0.591] |

For coverage-based selection, queries are ranked by $\rho(x) = m_{\widehat{Q}_\lambda}(x)/(2\varepsilon_{\text{fid}}(x) + 10^{-12})$, using the fitted winning margin defined in Appendix A.5. At each coverage $c \in \{0.1, 0.2, \ldots, 1.0\}$, the highest-ranked $\max\{1, \text{round}(cN)\}$ of the $N$ held-out queries are retained. The ranking is computed from the network and fitted scores. Held-out returns are used only to evaluate accuracy. Mean accuracy improves as coverage falls on Adult and Default. Bidding peaks at 0.763 with 20% retained and reaches 0.758 with 10% retained. The observed relationship supports selective use of the recovered structure, while a guarantee of task accuracy would require additional assumptions. The cross-task averages reported in the main text are arithmetic means of the three task-level seed means. At 10% coverage, gains over retaining all inputs are 8.70, 12.65, and 11.96 percentage points for PJM, Adult, and Default. Their mean is 11.10 points. The accuracy gap between inputs satisfying the condition and the remaining inputs averages 29.79 points, with the largest gap of 48.70 points on Adult. The network predictions remain fixed throughout selection, so the accuracy differences reflect which inputs are retained.

### E.5 Cross-seed stability of case coefficients

We hold training cases, returns, the train/test split, and 96 sampled queries fixed using seed 42, then vary network training randomness over seeds 42–46. Architectures, optimization settings, and ridge fitting match the three consistency tasks. The Default early-stopping split follows the network seed. For each query and each of the ten network pairs, we compare the signed coefficient vectors by Spearman rank correlation, identify the Top-30 cases by absolute coefficient, and measure coefficient-sign agreement on the union of those sets. Table E5 averages query results within pairs. Its standard deviations describe variation across the ten overlapping pairs, which are not independent replicates.

Table E5: Cross-seed case-coefficient stability. Entries are means $\pm$ descriptive standard deviations across ten network pairs. Rank correlation and coefficient-sign agreement exceed exact Top-30 set overlap. Chance is the approximate Jaccard overlap of independent uniform Top-30 sets.

| Task | Spearman | Sign agreement | Top-30 Jaccard | Chance |
|---|---|---|---|---|
| PJM | $0.488 \pm 0.049$ | $0.985 \pm 0.010$ | $0.174 \pm 0.024$ | 0.0011 |
| Adult | $0.354 \pm 0.036$ | $0.699 \pm 0.030$ | $0.043 \pm 0.005$ | 0.0006 |
| Default | $0.274 \pm 0.041$ | $0.683 \pm 0.033$ | $0.068 \pm 0.013$ | 0.0006 |

The chance reference is $k/(2n - k)$ for $k = 30$, the ratio of expected intersection to expected union sizes. The training sets contain 13,432 PJM, 24,129 Adult, and 24,000 Default cases. The observed overlaps exceed this reference, but their absolute values remain low. Coefficient direction and broad ranking are more stable than the identities of the largest coefficients. These comparisons concern representation-derived coefficients, while Appendix E.3 examines margin contributions under return perturbations. Together they show why the stability of a selected decision and the stability of its displayed cases should be evaluated separately.

### E.6 Confidence comparison and label-derived targets

We compare margin-to-error selection with ranking by the network's winning score minus its runner-up score. Both selectors use the same held-out inputs and coverage levels as Appendix E.4. Table E6 reports the trapezoidal area under each accuracy–coverage curve over coverage 0.1–1, divided by 0.9. Larger values indicate higher average retained accuracy. The confidence score performs better on Adult and PJM, while the paired interval for Default includes zero. The consistency condition provides a fitted–network agreement guarantee, but these comparisons do not establish an advantage in accuracy-based selection.

For Adult and Default, we additionally map the observed binary label to an action. Income above the Adult threshold maps to expanded allocation, and lower income to conservative allocation. Default maps to reducing the credit limit, and repayment to increasing it. The middle action is counted as a mismatch. Accuracies for inputs satisfying the condition and the remaining inputs are 0.836 [0.824, 0.849] and 0.331 [0.239, 0.424] for Adult, and 0.398 [0.377, 0.419] and 0.313 [0.230, 0.397] for Default. These targets test another reading of the same decisions. The labels also enter the

Table E6: Normalized area under the accuracy–coverage curve. The difference is margin-to-error selection minus confidence, with a paired 95% interval over five seeds.

| Task | Margin/error ratio | Confidence | Difference |
|---|---|---|---|
| PJM | 0.719 | 0.734 | $-0.015$ [$-0.025$, $-0.004$] |
| Adult | 0.884 | 0.922 | $-0.038$ [$-0.056$, $-0.020$] |
| Default | 0.796 | 0.789 | 0.007 [$-0.002$, 0.016] |

training-return construction, so this evaluation shares that dependence with the task-optimal accuracy analysis.

### E.7 Decision preservation with truncated case sums

For each seed and task, we sample 200 held-out queries and retain those satisfying the consistency condition. For each retained query, we order training cases by decreasing $|\alpha_{\lambda,i}(x)|$ and cumulatively sum their action-score contributions. We record the smallest prefix length $k$ after which every subsequent prefix selects the same action as the complete fitted score. This checks stability along the chosen ordering using all prefix decisions. It does not search all possible subsets.

Table E7: Case counts for stable truncated decisions on inputs satisfying the condition. Median and 90th percentile are computed within each seed and then averaged over five seeds. Brackets give 95% intervals for the seed medians. Large 90th percentiles indicate substantial variation across inputs.

| Task | Training cases | Mean median $k$ | Mean 90th percentile |
|---|---|---|---|
| PJM | 13,432 | 1.6 [0.9, 2.3] | 158.0 |
| Adult | 24,129 | 9.1 [5.9, 12.3] | 725.6 |
| Default | 24,000 | 23.9 [13.3, 34.5] | 1,647.6 |

Table E7 shows that some decisions satisfying the condition can be displayed using a small fraction of the training cases while retaining the fitted action. Other inputs require much longer case sums. These are measured truncation properties of the fitted scores. The consistency condition concerns the complete score and by itself does not guarantee a short case reconstruction.

## F Complementary Comparisons and Computational Cost

### F.1 Comparisons on the controlled margin reference

Table 2 compares the recovered case contributions from case decomposition with attribution-inspired scores computed over the same fixed representation and fitted linear return models. This comparison examines how the recovered case structure relates to existing training-data attribution formulations under the same fitted decision model. The comparison uses the action pair and reference in Eq. (B1). These implementations operate on the fitted linear model, so their required state and scoring formulas are specified below.

Let $K_i(x) = \phi(x)^\top \phi(x_i)$ and let $e_i(a, b)$ be the difference of fitted training residuals for actions $a$ and $b$:

$$e_i(a, b) = [\widehat{Q}_\lambda(x_i, a) - r_i(a)] - [\widehat{Q}_\lambda(x_i, b) - r_i(b)]. \tag{F1}$$

The case-decomposition score is $\chi_{\lambda,i}(x; a, b) = \alpha_{\lambda,i}(x)\Delta r_i(a, b)$, with $\lambda = 10^{-5}$. The influence-function score is $\phi(x)^\top H_d^{-1}\phi(x_i)e_i(a, b)$, where $H_d = (2/n)G + 10^{-3}I_H$. The representer-style score is $-K_i(x)e_i(a, b)/(2\lambda_{\text{rep}}n)$ with $\lambda_{\text{rep}} = 10^{-3}$. The single-checkpoint gradient score is $0.01K_i(x)e_i(a, b)$, and the inner-product score is $K_i(x)e_i(a, b)$.

These fixed-representation specializations are inspired by Influence Functions, representer points, and TracIn (Koh & Liang, 2017; Yeh et al., 2018; Pruthi et al., 2020), using the same ridge residuals and one final checkpoint. This comparison characterizes how recovered case contributions relate to

attribution-inspired scores for a fixed representation. The three inner-product scores have identical absolute rankings. The negative representer scale reverses their signs.

Table F1: Paired differences between case decomposition and fixed-representation comparison scores on the controlled margin reference. Differences are case decomposition minus the comparison score. All eight 95% intervals lie above zero. Two-sided paired $t$-tests use the five within-seed differences; $p$ values are unadjusted for multiple comparisons.

| Metric | Comparison score | Difference [95% CI] | Paired p |
|---|---|---|---|
| Pearson | Influence Functions | 0.623 [0.495, 0.751] | 0.0002 |
| Jaccard | Influence Functions | 0.266 [0.178, 0.355] | 0.0011 |
| nDCG | Influence Functions | 0.351 [0.227, 0.475] | 0.0014 |
| Sign agreement | Influence Functions | 0.330 [0.266, 0.394] | 0.0001 |
| Pearson | Representer-style | 0.574 [0.497, 0.650] | <0.0001 |
| Jaccard | Representer-style | 0.348 [0.235, 0.461] | 0.0010 |
| nDCG | Representer-style | 0.545 [0.351, 0.740] | 0.0015 |
| Sign agreement | Representer-style | 0.433 [0.375, 0.491] | <0.0001 |

The paired comparisons in Table F1 support closer alignment between the recovered contributions and the specified relevance-and-return reference. That reference evaluates case-level margin structure, whereas residual-based attribution scores target changes associated with training examples. The comparison therefore concerns their alignment with this reference under the same fitted model.

### F.2 Action consistency and case-removal refitting

The unweighted majority action among the same Top-30 margin-ranked cases agrees with the query's selected action at rates of 0.883 for Influence Functions, 0.865 for case decomposition, and 0.729 for the other two scores. This case-action statistic gives a different ordering from margin-reference alignment.

Removing the ten highest-ranked supporting cases and refitting gives case decomposition a normalized margin reduction of 0.0019 [−0.0044, 0.0082]. Its differences from random removal and representer-style removal are 0.0015 [−0.0048, 0.0078] and −0.0150 [−0.0366, 0.0065]. These intervals do not establish stronger removal effects. Refitting changes the Gram system and return table while retaining the trained representation. The decomposition describes contributions to the original fitted margin. The removal experiment evaluates their effect when refitting the readout with the learned representation held fixed.

### F.3 Computational cost

For a fixed representation, dense Gram preparation costs $O(nH^2 + H^3)$. After factorization, forming all query coefficients costs $O(H^2 + nH)$, followed by $O(n)$ to form one pairwise contribution vector. Case ranking adds sorting cost.

With 500 training cases and representation width 16, median cached query times are 0.0126 ms for case decomposition and Influence Functions, and approximately 0.0125 ms for the other two scoring rules. These measurements exclude training and representation evaluation, using 500 measured queries after 50 warm-up queries per seed. Including representation evaluation and Top-30 selection gives per-seed median times of 0.1135–0.1602 ms. Both measurements use an Intel Core i5-14500 central processing unit (CPU), Windows 11, and the Intel Math Kernel Library backend.