BEHAVIOR LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Interpretable machine learning is increasingly vital for scientific research, yet the performance–interpretability trade-off, insufficient alignment with scientific theory, and non-identifiability limit its scientific credibility. Grounded in behavioral science, we propose *Behavior Learning* (BL), a novel general-purpose ML framework that unifies predictive performance, intrinsic interpretability, and identifiability for scientifically credible modeling. BL discovers interpretable and identifiable optimization structures from data. It does so by parameterizing a compositional utility function built from intrinsically interpretable modular blocks, which induces a data distribution for prediction and generation. Each block represents and can be written in symbolic form as a utility maximization problem (UMP), a foundational paradigm in behavioral science and a universal framework of optimization. BL supports architectures ranging from a single UMP to hierarchical compositions, the latter modeling hierarchical optimization systems that offer both expressiveness and structural transparency. Its smooth and monotone variant (IBL) guarantees identifiability under mild conditions. Theoretically, we establish the universal approximation property of both BL and IBL, and analyze the M-estimation properties of IBL. Empirically, BL demonstrates strong predictive performance, intrinsic interpretability and scalability to high-dimensional data.

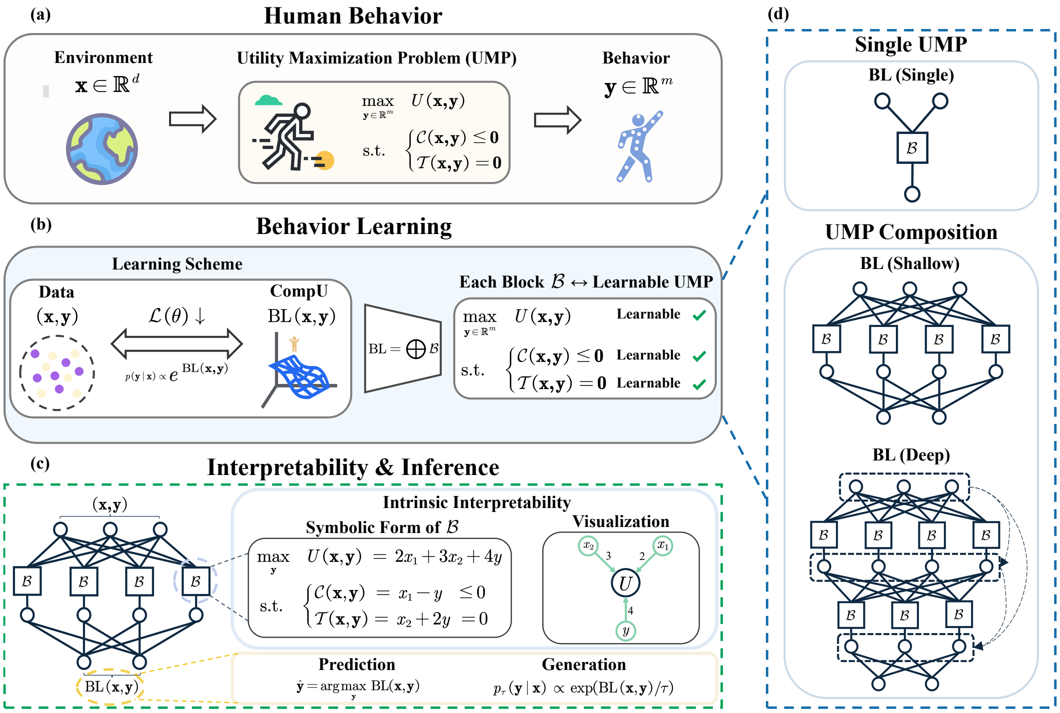
(a)
Human Behavior
Environment
$\mathbf{x} \in \mathbb{R}^d$
Utility Maximization Problem (UMP)
$\max_{\mathbf{y} \in \mathbb{R}^m} U(\mathbf{x},\mathbf{y})$
s.t. $\begin{cases} \mathcal{C}(\mathbf{x},\mathbf{y}) \leq \mathbf{0} \\ \mathcal{T}(\mathbf{x},\mathbf{y}) = \mathbf{0} \end{cases}$
Behavior
$\mathbf{y} \in \mathbb{R}^m$
(b)
Behavior Learning
Learning Scheme
Data
$(\mathbf{x},\mathbf{y})$
$\mathcal{L}(\theta) \downarrow$
$p(\mathbf{y} \mid \mathbf{x}) \propto e^{\mathrm{BL}(\mathbf{x},\mathbf{y})}$
CompU
$\mathrm{BL}(\mathbf{x},\mathbf{y})$
$\mathrm{BL} = \bigoplus \mathcal{B}$
Each Block $\mathcal{B} \leftrightarrow$ Learnable UMP
$\max_{\mathbf{y} \in \mathbb{R}^m} U(\mathbf{x},\mathbf{y})$ Learnable ✓
s.t. $\mathcal{C}(\mathbf{x},\mathbf{y}) \leq \mathbf{0}$ Learnable ✓
$\mathcal{T}(\mathbf{x},\mathbf{y}) = \mathbf{0}$ Learnable ✓
(c)
Interpretability & Inference
$(\mathbf{x},\mathbf{y})$
$\mathcal{B}$
$\mathrm{BL}(\mathbf{x},\mathbf{y})$
Intrinsic Interpretability
Symbolic Form of $\mathcal{B}$
$\max_{\mathbf{y}} \quad U(\mathbf{x},\mathbf{y}) = 2x_1 + 3x_2 + 4y$
s.t. $\begin{cases} \mathcal{C}(\mathbf{x},\mathbf{y}) = x_1 - y \leq 0 \\ \mathcal{T}(\mathbf{x},\mathbf{y}) = x_2 + 2y = 0 \end{cases}$
Visualization
x_2 3 2 x_1
U
4
y
Prediction
$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} \mathrm{BL}(\mathbf{x},\mathbf{y})$
Generation
$p_\tau(\mathbf{y} \mid \mathbf{x}) \propto \exp(\mathrm{BL}(\mathbf{x},\mathbf{y})/\tau)$
(d)
Single UMP
BL (Single)
$\mathcal{B}$
UMP Composition
BL (Shallow)
BL (Deep)

Figure 1: **Behavior Learning (BL).** (a) Human behavior modeled as a UMP. (b) Learning scheme of BL, where CompU denotes the compositional utility function. (c) BL offers intrinsic interpretability (via symbolic form as an optimization problem), identifiability (via unique parameterization), and inference capability. (d) Three architectural variants of BL, from single UMP to deep compositions.

1 Introduction

Scientific research often grapples with phenomena that resist precise formalization (Anderson, 1972; Mitchell, 2009), including human and social domains (Simon, 1955; Arthur, 2009). Such phenomena are difficult to predict and even harder to falsify through theory alone. Interpretable machine learning (Interpretable ML) (Molnar, 2020), with its powerful approximation capabilities and built-in transparency, offers a promising alternative for modeling such phenomena. Yet a long-standing tension remains unresolved: model predictive performance and intrinsic interpretability often trade off—a challenge commonly known as the *performance–interpretability trade-off* (Arrieta et al., 2020). High-performing models such as deep neural networks (LeCun et al., 2015) typically lack transparency, while intrinsically interpretable models struggle to capture complex nonlinear patterns.

Some efforts have been made to mitigate the performance–interpretability trade-off, and the main existing approaches can be broadly grouped into four categories. (i) Additive models (Caruana et al., 2015; Hastie, 2017; Nori et al., 2019; Agarwal et al., 2021; Chang et al., 2021). (ii) Concept-based models (Alvarez Melis & Jaakkola, 2018; Kim et al., 2018; Koh et al., 2020). (iii) Rule- and score-based systems (Ustun & Rudin, 2016; Angelino et al., 2018) . (iv) Shape-constrained neural networks (You et al., 2017) . Recent additional interpretable modeling frameworks include Kraus et al. (2024); Liu et al. (2024b); Plonsky et al. (2025). These approaches demonstrate varied strengths.

However, two fundamental limitations remain, restricting their scientific applicability. (i) *Insufficient alignment with scientific theories*. Most approaches focus on extending existing machine learning methods to achieve interpretability, rather than developing a scientifically grounded framework (e.g., based on optimization problems or differential equations). This often hinders alignment with scientific theories and limits the ability to extract scientific knowledge from learned models (Roscher et al., 2020; Bereska & Gavves, 2024; Longo et al., 2024). (ii) *Non-uniqueness of interpretations*. Most models are *non-identifiable*—their interpretations are not uniquely determined by observable predictions in a mathematical sense (Ran & Hu, 2017; Méloux et al., 2025). As a result, such models cannot support reliable estimation of ground-truth parameters (Newey & McFadden, 1994; Van der Vaart, 2000), and may even lack Popperian falsifiability (Popper, 2005), ultimately limiting their scientific credibility. These limitations naturally raise a key question: can we design an interpretable ML framework that mitigates the performance–interpretability trade-off while being scientifically grounded and identifiable?

Grounded in behavioral science, we propose **Behavior Learning (BL)**: *a novel, general-purpose interpretable ML framework for scientifically credible modeling*. It unifies high predictive performance, intrinsic interpretability, and identifiability. As illustrated in Figure 1, BL builds on one of the most fundamental paradigms in behavioral science—utility maximization—which posits that human behavior arises from solving a *utility maximization problem* (UMP) (Samuelson, 1948; Debreu, 1959; Mas-Colell et al., 1995). Motivated by this paradigm, BL learns interpretable latent optimization structures from data. It models responses (y) as drawn from a probability distribution induced by a UMP or a composition of multiple interacting UMPs. This distribution is parameterized by a compositional utility function $\mathrm{BL}(\mathbf{x}, \mathbf{y})$, constructed from intrinsically interpretable modular blocks $\mathcal{B}(\mathbf{x}, \mathbf{y})$. Each block is a learnable penalty-based formulation that represents a **optimization problem** (UMP), which can be written in symbolic form and offers transparency comparable to linear regression.

BL admits three architectural variants: *BL(Single)*, defined by a single block; *BL(Shallow)*, a moderately layered composition of blocks; and *BL(Deep)*, a deep hierarchical composition of multiple blocks. The latter two model, and can be symbolically interpreted as, **hierarchical optimization systems**. All variants are trained end-to-end to induce a conditional Gibbs distribution for prediction and generation. By refining the penalty functions in each block into smooth and monotone forms, we develop *Identifiable BL* (IBL), the identifiable variant of BL. Under mild conditions, IBL guarantees unique intrinsic interpretability. This property ensures the scientific credibility of its explanations and further supports recovery of the ground-truth model under appropriate conditions.

While motivated by behavioral science, *BL is not domain-specific*. It applies broadly to any scientific domain where observed outcomes arise as solutions to (explicit or latent) optimization problems—such as macroeconomics (Ramsey, 1928; Ljungqvist & Sargent, 2018), statistical physics (Gibbs, 1902; Landau & Lifshitz, 2013), or evolutionary biology (Wright et al., 1932; Fisher, 1999).

This generality is supported by a key theoretical insight (Theorem 2.2): any optimization problem can be equivalently written as a UMP. This makes BL a general-purpose modeling framework for *data-driven inverse optimization* (Ahuja & Orlin, 2001) across diverse scientific disciplines.

BL connects to three major research areas. (i) *Interpretable machine learning.* BL introduces a novel framework of interpretable ML that are optimization-grounded, symbolically expressible, and identifiable, thereby supporting scientifically credible modeling. (ii) *Inverse optimization.* BL relates to data-driven inverse optimization (Ahuja & Orlin, 2001; Keshavarz et al., 2011) and inverse reinforcement learning (Ng et al., 2000; Wulfmeier et al., 2015), but differs by learning the full constrained optimization structure, its associated training scheme, and the hierarchical compositions built upon it. (iii) *Energy-based models.* BL shares training techniques with energy-based models(LeCun et al., 2006), such as Gibbs-style modeling and denoising score matching (Hyvärinen & Dayan, 2005; Vincent, 2011). Instead of learning an opaque neural energy function, BL learns compositions of interpretable optimization problems.

We study BL both theoretically and empirically. Theoretically, we show that both BL and IBL admit universal approximation under mild assumptions (Section 2.2). For IBL, we further establish its M-estimation properties (Section 2.3), including identifiability, consistency, universal consistency, asymptotic normality, and asymptotic efficiency. Empirically, we evaluate BL across four tasks. Standard prediction tasks (Section 3.1) demonstrate its strong predictive performance. Counterfactual prediction (Section 3.2) highlights its potential applications in causal inference. A qualitative case study (Section 3.3) illustrates its intrinsic interpretability. Finally, prediction on high-dimensional inputs (Section 3.4) demonstrates its scalability to high-dimensional data. Due to limited space, we defer related works to Appendix A.

Overall, our key contributions are threefold. (i) We propose Behavior Learning (BL), a novel general-purpose machine learning framework grounded in behavioral science, which unifies high predictive performance, intrinsic interpretability, identifiability, and scalability. (ii) For scientific research, BL offers a scientifically grounded and identifiable interpretable ML approach for modeling complex phenomena that defy precise formalization. BL applies broadly to scientific disciplines associated with optimization. (iii) At the paradigm level, BL learns from data the optimization structure of either a single optimization problem or a hierarchical composition of problems through distributional modeling, contributing a new methodology to data-driven inverse optimization.

2 Behavior Learning (BL)

2.1 Utility Maximization Problem (UMP)

The modeling of human behavior, particularly in behavioral science and decision theory, often begins with the assumption that observed outcomes arise from a latent optimization process. A canonical formulation of this idea is the Utility Maximization Problem (UMP) (Mas-Colell et al., 1995), in which an agent selects actions $\mathbf{y} \in \mathcal{Y}$ in response to contextual features $\mathbf{x} \in \mathcal{X}$ by solving:

$$\max_{\mathbf{y}\in\mathcal{Y}} U(\mathbf{x}, \mathbf{y}) \quad \text{s.t.} \quad \mathcal{C}(\mathbf{x}, \mathbf{y}) \le 0, \quad \mathcal{T}(\mathbf{x}, \mathbf{y}) = 0 \tag{1}$$

Here, $U(\cdot)$ denotes a subjective utility function encoding the agent's internal preferences or goals. The inequality constraint $\mathcal{C}(\cdot)$ captures resource constraints, while the equality constraint $\mathcal{T}(\cdot)$ encodes either endogenous belief consistency or exogenous conservation laws.

The UMP can be recast as a cost–benefit framework, where the agent trades off utility gains against constraint violations. Formally, under mild regularity conditions, it admits an equivalent unconstrained reformulation via a penalty formulation (Han & Mangasarian, 1979), as formalized below.

Theorem 2.1 (Penalty Function Equivalence for UMP). *Let $\mathcal{X} \subset \mathbb{R}^{d_x}$ and $\mathcal{Y} \subset \mathbb{R}^{d_y}$ be nonempty compact sets, and let $U : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, $\mathcal{C} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^m$, and $\mathcal{T} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^p$ be Lipschitz continuous. Assume Slater's condition holds for the Utility Maximization Problem (UMP). Then there exist $\lambda_0 > 0$, $\lambda_1 \in \mathbb{R}^m_{++}$, $\lambda_2 \in \mathbb{R}^p_{++}$ such that the unconstrained objective*

$$\max_{\mathbf{y}\in\mathcal{Y}} \lambda_0\, \phi\big(U(\mathbf{x}, \mathbf{y})\big) - \lambda_1^\top \rho\big(\mathcal{C}(\mathbf{x}, \mathbf{y})\big) - \lambda_2^\top \psi\big(\mathcal{T}(\mathbf{x}, \mathbf{y})\big) \tag{2}$$

have the same global maximizers. Here, $\phi : \mathbb{R} \to \mathbb{R}$ is a strictly increasing C^1 map, and $\rho, \psi : \mathbb{R} \to \mathbb{R}_{\ge 0}$ are convex "penalty" functions satisfying $\rho(z) = 0$ for $z \le 0$, $\rho(z) > 0$ for $z > 0$; and $\psi(-z) = \psi(z)$, $\psi(0) = 0$, $\psi(z) > 0$ for $z \ne 0$.

The proof is provided in Appendix F.1. This unconstrained reformulation offers greater tractability for both theoretical analysis and model training.

While motivated by behavioral modeling, the UMP formulation is not domain-specific. It applies to any setting where observed outcomes are solutions to (explicit or latent) optimization problems. This is because any optimization problem can be equivalently formulated as a UMP. We state this in the following result, while the formal statement and proof are provided in Appendix F.1.

Theorem 2.2 (Universality of UMP)**.** *Any optimization problem of the form* $\max_{\mathbf{y}\in\mathcal{Y}} f(\mathbf{x},\mathbf{y})$ *or* $\min_{\mathbf{y}\in\mathcal{Y}} f(\mathbf{x},\mathbf{y})$*, subject to equality and inequality constraints, is equivalent to a UMP.*

2.2 BL Architecture

Figure 1(b–d) illustrates the architecture of BL. We consider samples $(\mathbf{x},\mathbf{y}) \sim \mathcal{D}$, where $\mathbf{x} \in \mathbb{R}^d$ denotes contextual features and $\mathbf{y}$ is the response, represented as $(\mathbf{y}^{\text{disc}}, \mathbf{y}^{\text{cont}}) \in \mathcal{Y}_{\text{disc}} \times \mathbb{R}^{m_c}$, capturing its hybrid structure. Responses are assumed to be stochastically generated by solving multiple interacting UMPs, each with a penalty-based formulation, which together compose a compositional utility function $\mathrm{BL}(\mathbf{x},\mathbf{y})$. On this basis, we model the data using a conditional Gibbs distribution (Gibbs, 1902) parameterized by $\mathrm{BL}_\Theta(\mathbf{x},\mathbf{y})$:

$$p_\tau(\mathbf{y} \mid \mathbf{x};\Theta) = \frac{\exp\big(\mathrm{BL}_\Theta(\mathbf{x},\mathbf{y})/\tau\big)}{Z_\tau(\mathbf{x};\Theta)}, \quad Z_\tau(\mathbf{x};\Theta) = \int_{\mathcal{Y}} \exp\big(\mathrm{BL}_\Theta(\mathbf{x},\mathbf{y}')/\tau\big)\, d\mathbf{y}' \tag{3}$$

Here the temperature parameter $\tau > 0$ controls the randomness of the response. As $\tau \to 0$, the distribution in equation 3 converges to a Dirac measure supported on $\arg\max_{\mathbf{y}} \mathrm{BL}(\mathbf{x},\mathbf{y})$, thereby recovering the deterministic best response obtained by solving the composed UMPs.

Model Structure of $\mathrm{BL}(\mathbf{x},\mathbf{y})$**.** To represent the composition of multiple UMPs, we build $\mathrm{BL}(\mathbf{x},\mathbf{y})$ by composing fundamental modular blocks $\mathcal{B}(\mathbf{x},\mathbf{y})$. Each block provides a penalty-based formulation of a single UMP, and together they yield the overall compositional utility function. Motivated by Theorem 2.1, we parameterize $\mathcal{B}(\mathbf{x},\mathbf{y})$ as

$$\mathcal{B}(\mathbf{x},\mathbf{y};\theta) := \lambda_0^\top \phi\big(U_{\theta_U}(\mathbf{x},\mathbf{y})\big) - \lambda_1^\top \rho\big(\mathcal{C}_{\theta_C}(\mathbf{x},\mathbf{y})\big) - \lambda_2^\top \psi\big(\mathcal{T}_{\theta_T}(\mathbf{x},\mathbf{y})\big) \tag{4}$$

where $\theta := (\lambda_0, \lambda_1, \lambda_2, \theta_U, \theta_C, \theta_T)$ denotes the complete set of learnable parameters. Following Theorem 2.1, ϕ is an increasing function; ρ penalizes inequality violations; and ψ captures symmetric deviations. Each block can be written as a well-defined UMP.

We then compose $\mathrm{BL}(\mathbf{x},\mathbf{y})$ from multiple $\mathcal{B}$-blocks in three structural forms to improve its representational power for optimization structures, as illustrated in Figure 1(d).

1. BL(Single) applies a single instance of $\mathcal{B}(\mathbf{x},\mathbf{y})$ as defined in equation 4, without any additional layers. It can be viewed as learning a single UMP, and offers maximal interpretability.
2. BL(Shallow) uses $\mathcal{B}(\mathbf{x},\mathbf{y})$ as the fundamental modular block to construct a shallow network. It introduces one or two intermediate layers of computation. Each layer $\mathbb{B}_\ell$ stacks multiple parallel $\mathcal{B}_{\ell,i}$ blocks to produce a vector in $\mathbb{R}^{d_\ell}$, i.e., $\mathbb{B}_\ell(\mathbf{x},\mathbf{y};\theta_\ell) := [\mathcal{B}_{\ell,1}(\mathbf{x},\mathbf{y};\theta_{\ell,1}), \ldots, \mathcal{B}_{\ell,d_\ell}(\mathbf{x},\mathbf{y};\theta_{\ell,d_\ell})]^\top$. The output of $\mathbb{B}_\ell$ is directly fed into the next $\mathbb{B}_{\ell+1}$, and only the final output is passed through a learnable affine transformation.
3. BL(Deep) extends the BL(Shallow) architecture to more than two layers, enabling richer hierarchical compositions of UMPs while maintaining the same recursive structure. As before, only the final output is affine transformed.

The overall structure of BL(Shallow) and BL(Deep) can be expressed in a unified form, where the shallow case corresponds to $L \le 2$ and the deep case to $L > 2$:

$$\mathrm{BL}(\mathbf{x},\mathbf{y}) := \mathbf{W}_L \cdot \mathbb{B}_L\big(\cdots \mathbb{B}_2(\mathbb{B}_1(\mathbf{x},\mathbf{y}))\cdots\big) \tag{5}$$

Learning Objective. The response $\mathbf{y}$ may contain both discrete and continuous components. For discrete responses, we directly apply cross-entropy (Kullback & Leibler, 1951) on $\mathbf{y}^{\text{disc}}$. For continuous responses, since the compositional utility function is analogous to an energy function (LeCun et al., 2006), we employ denoising score matching (Vincent, 2011) on $\mathbf{y}^{\text{cont}}$. The final objective combines the two with nonnegative weights $\gamma_{\text{d}}, \gamma_{\text{c}}$:

$$\mathcal{L}(\theta) = \gamma_{\text{d}}\, \mathbb{E}\big[-\log p_\tau(\mathbf{y}^{\text{disc}} \mid \mathbf{x})\big] + \gamma_{\text{c}}\, \mathbb{E}\, \big\|\nabla_{\tilde{\mathbf{y}}^{\text{cont}}} \log p_\tau(\tilde{\mathbf{y}}^{\text{cont}} \mid \mathbf{x}) + \sigma^{-2}(\tilde{\mathbf{y}}^{\text{cont}} - \mathbf{y}^{\text{cont}})\big\|^2 \tag{6}$$

Implementation Details. Here, we describe the key implementation choices for the general form of BL, taken as defaults unless otherwise noted. Further details are provided in Appendix E.3.

- Function Instantiation. Following equation 4, we instantiate the function $\mathcal{B}(\mathbf{x}, \mathbf{y})$ as
$$\mathcal{B}(\mathbf{x}, \mathbf{y}) = \lambda_0^\top \tanh\big(\mathbf{p}_u(\mathbf{x}, \mathbf{y})\big) - \lambda_1^\top \mathrm{ReLU}\big(\mathbf{p}_c(\mathbf{x}, \mathbf{y})\big) - \lambda_2^\top \big|\mathbf{p}_t(\mathbf{x}, \mathbf{y})\big| \tag{7}$$
where $\mathbf{p}_u, \mathbf{p}_c, \mathbf{p}_t$ are polynomial feature maps of bounded degree, providing interpretable representations of utility, inequality, and equality terms, respectively. The bounded $\tanh$ reflects the principle of diminishing marginal utility (Jevons, 2013), a commonly assumed principle in behavioral science, while ReLU and $|\cdot|$ introduce soft penalties for constraint violations.

- Polynomial Maps. In BL(Single), the structure of polynomial maps is optional. In BL(Shallow) and BL(Deep), each $\mathcal{B}$-block employs affine transformations as its polynomial maps, with higher-degree and interaction terms omitted by default for computational efficiency.

- Skip Connections. For deep variants, skip connections can be optionally introduced to improve representational efficiency.

Theoretical Guarantees. Under the given architecture, the BL framework has universal approximation power: it can approximate any continuous conditional distribution arbitrarily well, provided that BL has sufficient capacity, as stated below. The proof is given in Appendix F.2.

Theorem 2.3 (Universal Approximation of BL)**.** *Let $\mathcal{X} \subset \mathbb{R}^d$ and $\mathcal{Y} \subset \mathbb{R}^m$ be compact sets, and let $p^\star(\mathbf{y} \mid \mathbf{x})$ be any continuous conditional density such that $p^\star(\mathbf{y} \mid \mathbf{x}) > 0$ for all $(\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y}$. Then for any $\tau > 0$ and $\varepsilon > 0$, there exists a finite BL architecture (with depth and width depending on ε) and a parameter $\theta^\star$ such that the Gibbs distribution in equation 18 satisfies*
$$\sup_{\mathbf{x} \in \mathcal{X}} \mathrm{KL}\big(p^\star(\cdot \mid \mathbf{x}) \,\|\, p_\tau(\cdot \mid \mathbf{x}; \theta^\star)\big) < \varepsilon. \tag{8}$$

Interpretability. Alongside its expressive power, BL also exhibits strong intrinsic interpretability. (i) Each $\mathcal{B}$-block can be expressed in **symbolic form** as an optimization problem (UMP): the $\tanh$ term defines the objective, the ReLU term corresponds to an inequality constraint, and the absolute-value term corresponds to an equality constraint. Thus, BL(Single) can be directly expressed as a symbolic UMP, whereas deeper architectures can be interpreted as compositions of UMPs, with each block retaining interpretability. (ii) The **polynomial basis** ensures a level of **transparency comparable to linear regression**, as both objectives and constraints can be represented as linear combinations of polynomial features. It can further be visualized as a computational graph (Figure 6), in which each input's influence on every $\mathcal{B}$-block is traceable through compositional pathways. (iii) BL(Deep) composes $\mathcal{B}$-blocks in a layered manner, forming a **hierarchical optimization system**. Interpretation proceeds in a bottom-up fashion, where the relation between any two consecutive layers can be viewed as aggregation or coarse-grained observation. Overall, the interpretive pathway is: *raw input features → micro-level optimization blocks → macro-level aggregation or coarse-grained behavioral constructs → macro-level optimization systems*. Appendix B provides a detailed description of this interpretation procedure. (iv) BL also offers multiple architectural degrees of freedom that provide flexibility but simultaneously affect the resulting interpretability. In deep variants, skip connections introduce cross-layer dependency structures that are modeled in statistical physics (Yang & Schoenholz, 2017). Replacing polynomial maps with affine transformations preserves the underlying optimization semantics but reduces symbolic granularity, yielding a more qualitative rather than symbolic interpretation of each block. (v) BL can be interpreted as a single UMP when the final layer contains only one $\mathcal{B}$-block, since all lower-layer structures aggregate into a unified optimization problem. When the final layer contains multiple $\mathcal{B}$-blocks, BL corresponds to a linear trade-off among multiple optimization problems

2.3 Identifiable Behavior Learning (IBL)

Beyond prediction and interpretability, the BL framework supports a third fundamental goal: *the identification of ground-truth parameters*, which in turn endows BL with the capacity for scientifically credible modeling. We refer to this setting as **Identifiable Behavior Learning (IBL)**. In the IBL setting, we define the modular block as
$$\mathcal{B}^{\mathrm{id}}(\mathbf{x}, \mathbf{y}; \theta) := \lambda_0^\top \phi^{\mathrm{id}}\big(U_{\theta_U}(\mathbf{x}, \mathbf{y})\big) - \lambda_1^\top \rho^{\mathrm{id}}\big(\mathcal{C}_{\theta_C}(\mathbf{x}, \mathbf{y})\big) - \lambda_2^\top \psi^{\mathrm{id}}\big(\mathcal{T}_{\theta_T}(\mathbf{x}, \mathbf{y})\big) \tag{9}$$

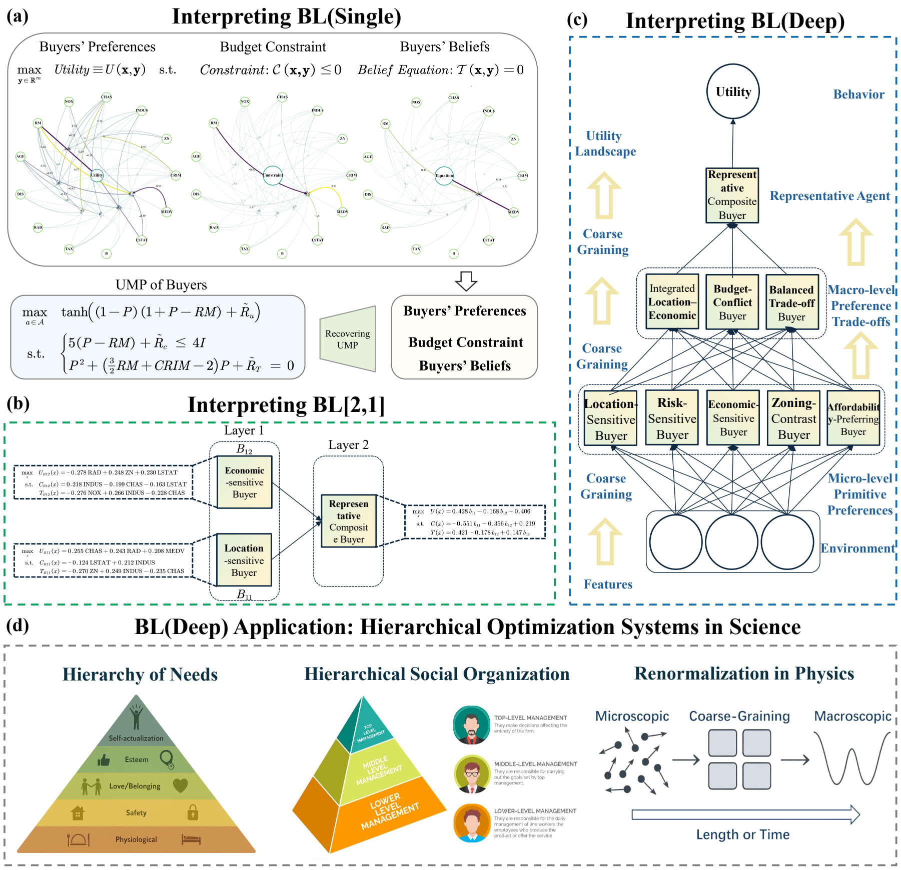
(a)
Interpreting BL(Single)
Buyers' Preferences
Budget Constraint
Buyers' Beliefs
$\max_{\mathbf{y}\in\mathbb{R}^m}$ Utility $\equiv U(\mathbf{x},\mathbf{y})$ s.t.
Constraint: $\mathcal{C}(\mathbf{x},\mathbf{y}) \le 0$
Belief Equation: $\mathcal{T}(\mathbf{x},\mathbf{y}) = 0$
CHAS
INDUS
NOX
RM
ZN
AGE
CRIM
DIS
MEDV
RAD
LSTAT
TAX
B
Utility
Constraint
Equation
UMP of Buyers
$\max_{a\in\mathcal{A}}$ $\tanh\big((1-P)(1+P-RM)+\tilde{R}_u\big)$
s.t. $5(P-RM)+\tilde{R}_c \le 4I$
$P^2+\big(\frac{3}{2}RM+CRIM-2\big)P+\tilde{R}_T = 0$
Recovering UMP
Buyers' Preferences
Budget Constraint
Buyers' Beliefs
(b)
Interpreting BL[2,1]
Layer 1
Layer 2
B_{12}
B_{11}
$\max_x$ $U_{B12}(x) = -0.278$ RAD $+ 0.248$ ZN $+ 0.230$ LSTAT
s.t. $C_{B12}(x) = 0.218$ INDUS $- 0.199$ CHAS $- 0.163$ LSTAT
$T_{B12}(x) = -0.276$ NOX $+ 0.266$ INDUS $- 0.228$ CHAS
$\max_x$ $U_{B11}(x) = 0.255$ CHAS $+ 0.243$ RAD $+ 0.208$ MEDV
s.t. $C_{B11}(x) = -0.124$ LSTAT $+ 0.212$ INDUS
$T_{B11}(x) = -0.270$ ZN $+ 0.249$ INDUS $- 0.235$ CHAS
Economic -sensitive Buyer
Location -sensitive Buyer
Represen tative Composit e Buyer
$\max_x$ $U(x) = 0.428\,b_{11} - 0.168\,b_{12} + 0.406$
s.t. $C(x) = -0.551\,b_{11} - 0.356\,b_{12} + 0.219$
$T(x) = 0.421 - 0.178\,b_{12} + 0.147\,b_{11}$
(c)
Interpreting BL(Deep)
Utility
Behavior
Utility Landscape
Represent ative Composite Buyer
Representative Agent
Coarse Graining
Integrated Location–Economic
Budget-Conflict Buyer
Balanced Trade-off Buyer
Macro-level Preference Trade-offs
Coarse Graining
Location-Sensitive Buyer
Risk-Sensitive Buyer
Economic-Sensitive Buyer
Zoning-Contrast Buyer
Affordability-Preferring Buyer
Coarse Graining
Micro-level Primitive Preferences
Environment
Features
(d)
BL(Deep) Application: Hierarchical Optimization Systems in Science
Hierarchy of Needs
Self-actualization
Esteem
Love/Belonging
Safety
Physiological
Hierarchical Social Organization
TOP-LEVEL MANAGEMENT
MIDDLE LEVEL MANAGEMENT
LOWER LEVEL MANAGEMENT
Renormalization in Physics
Microscopic
Coarse-Graining
Macroscopic
Length or Time

Figure 2: **(a)** Visualization and symbolic form of BL(Single) trained on the *Boston Housing* dataset, modeling the UMP ($\max U$ s.t. $\mathcal{C} \le 0$, $\mathcal{T} = 0$) of a representative buyer in Boston housing (details in Section 3.3). Top: computational graphs of the polynomials inside the three penalty functions—$\tanh$ (preference), ReLU (budget), and $|\cdot|$ (belief). Each graph is respectively centered on $\tanh^{-1}(U)$, $\mathcal{C}$, and $\mathcal{T}$ from left to right, with surrounding nodes representing input features. Directed edges (shown only if coefficient ≥ 0.3) indicate how each feature contributes to the corresponding term. Bottom: approximate symbolic formulation of the trained BL model as a UMP. **(b)** The BL[2,1] architecture. Layer 1 identifies two key micro-level preference types: the *Economic-sensitive Buyer* and the *Location-sensitive Buyer*. Layer 2 aggregates these two components into an effective representative buyer. **(c)** The BL(Deep) [5,3,1] architecture. Layer 1 recovers five distinct micro-level housing preference types. Layer 2 identifies three macro-level trade-off types capturing different ways these primitive preferences interact. Layer 3 aggregates them into the overall representative buyer. Table 11 provides detailed descriptions of each type. BL(Deep) provides a hierarchical explanation consistent with the coarse-graining principle (Kadanoff, 1966) in statistical physics, reconstructing the full micro-to-macro optimization hierarchy. In addition, the preference and trade-off patterns uncovered by BL(Deep) are well documented in the classical economics literature (see Table 12). **(d)** BL can be applied to a broad class of hierarchical optimization systems in science, including hierarchical need structures, hierarchical social–organizational systems, and renormalization-style coarse-grained systems in physics.

Unlike BL, which uses general nonlinearities, the IBL architecture imposes stricter structural constraints: ϕ^{id} and ρ^{id} are strictly increasing, while ψ^{id} is symmetric and strictly increasing in $|\cdot|$. In addition, all three functions are C^1. These properties ensure that each UMP block stays responsive and adjusts smoothly to objectives and constraints. In practice, we instantiate equation 9 as

$$\mathcal{B}^{\text{id}}(\mathbf{x},\mathbf{y}) = \lambda_0^\top \tanh\big(\mathbf{p}_u(\mathbf{x},\mathbf{y})\big) - \lambda_1^\top \operatorname{softplus}\big(\mathbf{p}_c(\mathbf{x},\mathbf{y})\big) - \lambda_2^\top \big(\mathbf{p}_t(\mathbf{x},\mathbf{y})\big)^{\odot 2} \tag{10}$$

where $(\cdot)^{\odot 2}$ denotes elementwise square.

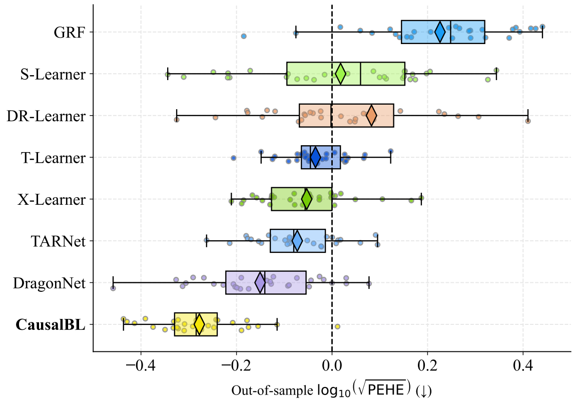

Figure 3: Counterfactual prediction performance (synthetic dataset). *IBL-based model significantly outperforms models based on NN, trees, and regression.*

Table 1: Within- and out-of-sample mean ± std of $\sqrt{\text{PEHE}}$ on the synthetic dataset. Top two per column are highlighted in blue and red. The dataset is *synthetically generated via nonlinear mathematical simulations with noise* (see Appendix H.3.1).

Model	Synthetic Dataset	
	Within-sample	Out-of-sample
IBL-based model	0.53 ± 0.09	0.54 ± 0.13
DR-Learner	2.11 ± 3.52	1.89 ± 3.03
DragonNet	0.82 ± 0.22	0.73 ± 0.20
GRF	1.91 ± 0.65	1.77 ± 0.54
S-Learner	1.22 ± 0.41	1.13 ± 0.45
TARNet	0.92 ± 0.21	0.86 ± 0.17
T-Learner	0.94 ± 0.13	0.93 ± 0.15
X-Learner	0.94 ± 0.14	0.91 ± 0.22

We design IBL in three architectural forms. Similar to BL, the *IBL(Single)* directly uses $\mathcal{B}^{\text{id}}(\mathbf{x}, \mathbf{y})$ as the compositional utility function. The *IBL(Shallow)* and *IBL(Deep)* variants are defined recursively as

$$\text{IBL}(\mathbf{x}, \mathbf{y}) := \mathbf{W}_L^{\circ} \cdot \mathbb{B}_L^{\text{id}}\big(\cdots \mathbb{B}_2^{\text{id}}\big(\mathbb{B}_1^{\text{id}}(\mathbf{x}, \mathbf{y})\big) \cdots\big), \quad L \geq 1 \tag{11}$$

where $\mathbb{B}_\ell^{\text{id}}$ stacks multiple parallel blocks $\mathcal{B}_{\ell,i}^{\text{id}}(\mathbf{x}, \mathbf{y})$, and $\mathbf{W}_L^{\circ}$ is a learnable affine transformation without bias. All other design choices follow the BL setting.

Theoretical Foundation. IBL admits favorable properties for ground-truth identification. We begin by establishing identifiability, which is fundamental for statistical inference. We first state our key assumption (see Assumption F.1 for details).

Assumption 2.1. *Let $\bar{\Psi}$ denote the quotient space of atomic parameters. We assume that the map $\bar{\Psi} \to \mathbb{R}^{\mathcal{X}\times\mathcal{Y}}$, $\bar{\psi} \mapsto g_{\bar{\psi}}$, is **injective**, and that any finite set of distinct atoms is **linearly independent**. We further restrict attention to **minimal representations** with no duplicate atoms and a fixed canonical ordering.*

Theorem 2.4 (Identifiability of IBL)**.** *Under Assumption F.1, the architectures IBL(Single), IBL(Shallow), and IBL(Deep) are identifiable in the parameter quotient space $\bar{\Theta}$.*

Theorem 2.5 (Loss Identifiability of IBL)**.** *The IBL model is parameterized by $\theta \in \Theta$. Suppose Θ is compact. Then under Assumption F.1, the population loss $\mathcal{L}$ defined in equation 6 satisfies:*

- *If $\gamma_c > 0$, it admits a **unique** minimizer in the quotient space $\bar{\Theta}$;*
- *If $\gamma_c = 0$, it admits a **unique** minimizer in the scale-invariant quotient space $\widetilde{\Theta}$.*

Theorems 2.4 and 2.5 together establish the identifiability of IBL. Theorem 2.4 shows that if two IBL models of the same structure induce the same compositional utility, then their parameters coincide up to an equivalence class. Theorem 2.5 further extends this result to loss-based identifiability. These results jointly imply that IBL admits a unique parameter estimate up to an equivalence class, and thus yields intrinsic interpretability that is unique up to the same class.

Building on identifiability, Theorem 2.6 establishes the statistical consistency of IBL: under compactness of the parameter space, the learned parameters converge in probability to a minimizer of the population loss as the sample size $n \to \infty$. If the model is correctly specified, the estimator further *converges to the ground-truth parameter, recovering the true underlying model*, thereby endowing IBL with the potential to recover the ground-truth model.

Theorem 2.6 (Consistency of IBL)**.** *Let Ξ denote the relevant parameter quotient space: $\Xi = \bar{\Theta}$ if $\gamma_c > 0$, and $\Xi = \widetilde{\Theta}$ if $\gamma_c = 0$. Let $\hat{\theta}_n \in \arg\min_{\theta\in\Theta} \mathcal{M}_n(\theta)$ denote the empirical minimizer, and let $\theta^{\bullet} \in \arg\min_{\theta\in\Theta} \mathcal{M}(\theta)$ denote the population minimizer. Then under the conditions of Theorem F.5,*

$$\hat{\theta}_n \xrightarrow{p} \theta^{\bullet} \quad \textit{in } \Xi, \qquad \mathcal{M}(\hat{\theta}_n) \xrightarrow{p} \mathcal{M}(\theta^{\bullet}).$$

Moreover, if the model is correctly specified (i.e., the data distribution is realized by some $\theta^\star \in \Theta$), then $\theta^\bullet = \theta^\star$ in Ξ, and thus $\hat{\theta}_n \xrightarrow{p} \theta^\star$.

Correct specification is a strong and often unrealistic assumption. Fortunately, the IBL framework—like BL—also enjoys a **universal approximation guarantee** (Theorem F.6). Building on this result, we further establish the **universal consistency** of IBL: *even under misspecification, IBL is capable of recovering the ground-truth model* with sufficiently large sample sizes.

Theorem 2.7 (Universal Consistency of IBL)**.** *Under the conditions of Theorem F.7, for any admissible data-generating distribution $p^\dagger$ satisfying the regularity assumptions of Theorem F.6, the IBL posterior sequence $\{p_{\hat{\theta}_n}\}$ satisfies*

$$\sup_{x \in \mathcal{X}} \mathrm{KL}\big(p^\dagger(\cdot \mid \mathbf{x}) \,\|\, p_{\hat{\theta}_n}(\cdot \mid \mathbf{x})\big) \xrightarrow{p} 0,$$

i.e., the learned conditional distributions $\{p_{\hat{\theta}_n}\}$ converge in KL to $p^\dagger$ uniformly over $\mathbf{x}$.

Specifically, this result implies that, even under model misspecification, the learned predictive distribution $p_{\hat{\theta}_n}$, parameterized by the IBL model, converges uniformly in KL to the true conditional distribution $p^\dagger$, provided that the capacity of the IBL architecture grows with the sample size n.

We also establish the **asymptotic normality** of IBL estimators (Theorem F.9), showing that the parameter estimates converge in distribution to a normal law as the sample size increases. Furthermore, under additional regularity conditions, the asymptotic variance attains the **efficient information bound** (Theorem F.10), demonstrating the statistical optimality of IBL.

Formal statements and proofs of all theorems in this part are deferred to Appendix F.3.

3 Experiments

In this section, we conduct four groups of experiments to systematically evaluate the capabilities of BL. Due to space constraints, details are provided in Appendix H.

3.1 Standard Prediction Tasks

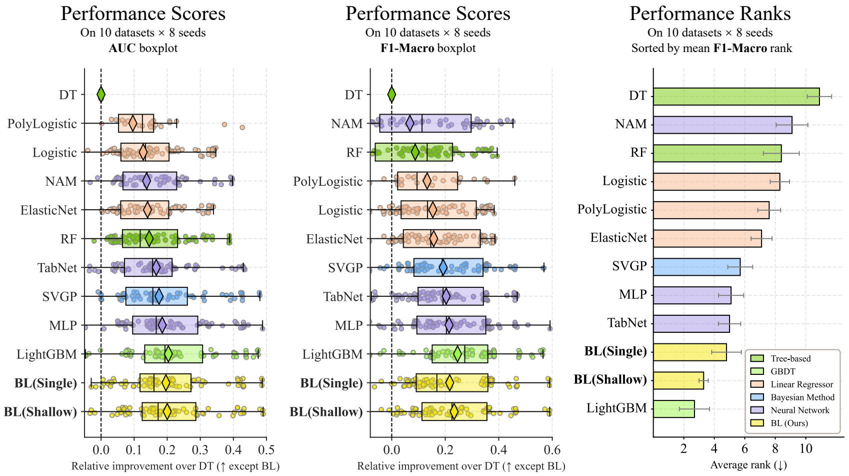

Figure 4: Predictive performance of BL and baselines. Left/Middle: relative AUC and F1-Macro gains over DT, sorted by mean (excluding BL). Right: mean F1-Macro ranks (↓ better). BL achieves first-tier performance in both metrics. Its variants rank second and third in mean F1-Macro rank, with BL(Shallow) showing no statistically significant difference from state-of-the-art models.

Is BL accurate enough for standard prediction tasks? In this part, we evaluate the predictive performance of BL on *10 datasets* (Table 3), covering diverse sample sizes, feature dimensions, and scientific domains. For fair comparison, we consider two BL variants—BL(Single) and

BL(Shallow)—and compare them against *10 baseline models* (Table 4) drawn from five methodological families: neural networks, tree-based models, gradient boosting methods, Bayesian methods, and linear regressors. All methods share a unified preprocessing and tuning pipeline.

Predictive Performance. Figure 4 shows that BL attains first-tier predictive performance overall, achieving the best results among intrinsically interpretable models. Notably, BL(Shallow) surpasses MLP, highlighting that BL delivers interpretability without sacrificing performance.

3.2 Counterfactual Prediction

Does BL exhibit potential for causal inference? To investigate this, we propose a causal extension of IBL designed to estimate individual treatment effects (ITEs), with details provided in Appendix G. We evaluate IBL-based model on *three datasets*: a synthetic dataset with known individual treatment effects (ITEs), the semi-synthetic IHDP-100 (Hill, 2011), and the real-world Jobs dataset (LaLonde, 1986). The synthetic dataset follows de Vassimon Manela et al. (2024), with added covariate nonlinearity and stochastic noise to increase complexity (Appendix H.3.1), and includes 30 replications; IHDP and Jobs contain 100 and 10 realizations, respectively. IBL-based model is compared against *seven widely used baselines* (Table 8) spanning four methodological families: tree-based models, representation-learning networks, meta-learners, and doubly robust methods.

Counterfactual Prediction Performance. Figure 3, Table 1 and Table 9 report counterfactual prediction results on three datasets. On the synthetic dataset, IBL-based model consistently outperforms all baselines with the lowest variance, demonstrating its strong capacity to model complex nonlinear structures. On IHDP, IBL-based model achieves the second-lowest $\sqrt{\text{PEHE}}$ and ATE relative error, while on Jobs it attains the lowest $|\text{ATT}|$ error. Notably, **IBL-based model is the only intrinsically interpretable model** among all competitors.

3.3 Interpreting BL: A Case Study

How can BL be interpreted in practice? This part presents a case study using the *Boston Housing* dataset, where we train a supervised BL(Single) model with a degree-2 polynomial basis, a BL[2,1] model (i.e., a two-layer BL with two B-blocks in the first layer and one in the second layer), and a BL(Deep) model with a [5,3,1] architecture to predict median home values. We illustrate how the internal structure of BL can be interpreted as explicit optimization problems and their hierarchical versions, accompanied by complementary visualizations. Further details are provided in Appendix H.4 and H.9.

Symbolic Form of BL(Single) as a UMP. As shown in Figure 2, the trained BL(Single) model can be interpreted as the UMP of a *representative buyer* in the Boston Housing market, comprising a single objective, inequality, and equality term. Each term is represented by an estimated quadratic polynomial. For parsimony, we extract approximate symbolic expressions by retaining only the monomials with the largest (2–5) absolute coefficients, while collecting the remaining terms (including constants) into a residual term $\tilde{R}$. For example, the utility term can be written as:

$$\mathbf{p}_u = -0.56 \cdot P^2 - 0.6 \cdot \text{RM} + 0.57 \cdot \text{RM} \cdot P + \tilde{R}_u \approx (1 - P)(1 + P - \text{RM}) + \tilde{R}_u$$

We similarly simplify the budget and belief terms to recover an approximate UMP for the buyer. The full symbolic form is illustrated at the bottom of Figure 2.

Interpreting BL(Single) via Model Visualization. Visualizations of each term's polynomial reveal how features constitute the UMP. Three insights emerge from the visualizations in Figure 2. (i) *Median housing price (MEDV)* and *average number of rooms (RM)* are dominant across all terms—MEDV negatively affects utility in a near-quadratic form, while RM modulates its marginal effect. (ii) *Proportion of lower-income residents (LSTAT)* features prominently in the budget constraint, reflecting implicit resource limitations. (iii) *Crime rate (CRIM)* appears only in the belief term, suggesting that buyers treat it as influencing others' behavior rather than their own preferences.

Interpreting BL(Deep). (1) Figure 2 (b) illustrates the optimization problems learned by the BL[2,1] model. Layer 1 identifies two micro-level preference types: an *Economic-sensitive Buyer*, whose utility and constraint terms load primarily on ZN (Large-lot residential share) and LSTAT (Proportion of lower-income residents); and a *Location-sensitive Buyer*, driven mainly by CHAS (Charles River indicator) and RAD (Highway accessibility). Layer 2 aggregates these basic preferences, yielding an effective "representative buyer" that integrates the two preference types. (2) Figure 2 (c) presents the internal structure of the BL[5,3,1] model. In Layer 1, BL recovers five distinct *micro-level preference* types characterizing heterogeneous patterns in the housing market. Layer 2 identifies three macro-level representative agents, each capturing a different *macro-level trade-off* among the basic preferences. Layer 3 then aggregates these components into a single high-level mechanism, yielding the overall representative buyer. Table 11 provides detailed descriptions of each type. (3) Beyond interpretability, we find that **each preference pattern and trade-off recovered by BL(Deep) aligns with established findings in the economics literature** (see Table 12). This indicates that BL successfully reconstructs underlying scientific knowledge.

3.4 Prediction on High-Dimensional Inputs

Is BL scalable to high-dimensional inputs? We evaluate BL against the *energy-based MLP* (E-MLP) baseline across network depths $d \in \{1, 2, 3\}$, with all models implemented without skip connections. Experiments are conducted on *four datasets* spanning both image and text domains, and are evaluated using *six metrics*: in-distribution accuracy, calibration metrics (ECE and NLL), and OOD robustness metrics (AUROC, AUPR, and FPR@95). For OOD evaluation, we adopt symmetric ID↔OOD splits, using MNIST (LeCun et al., 2002) and Fashion-MNIST (Xiao et al., 2017) as one pair, and AG News and Yelp Polarity (Zhang et al., 2015) as another. E-MLP and BL are controlled to have comparable parameters.

Scalability on High-Dimensional Inputs. Figure 5 and Table 2 present results for BL and E-MLP across network depths. Overall, the two models exhibit comparable ID accuracy and OOD AUROC across datasets. On both the Fashion-MNIST and AGNews datasets, however, BL generally achieves higher OOD AUROC than E-MLP at similar accuracy levels. This indicates its stronger out-of-distribution generalization and robustness. BL also achieves better ECE and NLL (Table 19).

Downward Shift of the Pareto Frontier. Table 14 reports the parameter counts of BL and E-MLP across four tasks, and Tables 15-18 summarize their runtimes. The two models have highly comparable parameter sizes. Notably, BL runs substantially faster than E-MLP on text datasets, while being slightly slower on image datasets. These results indicate that BL achieves a *downward shift* of the Pareto frontier.

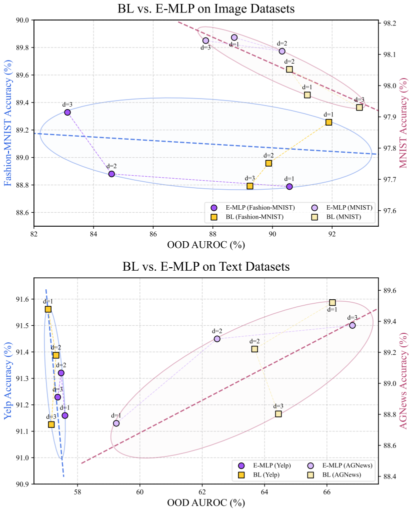

Figure 5: Comparison of BL and E-MLP on image and text datasets; d denotes model depth.

Table 2: ID accuracy and OOD AUROC (%) on image and text datasets. BL and E-MLP are evaluated at depths 1–3 with matched parameter counts, both without skip connections. Top-two per column are blue and red.

Model	Image Datasets			
	MNIST		Fashion-MNIST	
	Accuracy	OOD AUROC	Accuracy	OOD AUROC
E-MLP (depth=1)	98.15 ± 0.07	88.72 ± 1.36	88.79 ± 0.29	90.57 ± 1.39
BL (depth=1)	97.97 ± 0.18	91.17 ± 2.68	89.26 ± 0.22	91.89 ± 0.71
E-MLP (depth=2)	98.11 ± 0.08	90.32 ± 1.74	88.88 ± 0.26	84.61 ± 2.56
BL (depth=2)	98.05 ± 0.12	90.57 ± 2.49	88.96 ± 0.39	89.87 ± 2.48
E-MLP (depth=3)	98.14 ± 0.11	87.76 ± 2.55	89.33 ± 0.25	83.13 ± 1.90
BL (depth=3)	97.93 ± 0.27	92.92 ± 1.69	88.79 ± 0.25	89.24 ± 4.18
Model	Text Datasets			
	AG News		Yelp	
	Accuracy	OOD AUROC	Accuracy	OOD AUROC
E-MLP (depth=1)	88.74 ± 0.26	59.24 ± 0.21	91.16 ± 0.02	57.60 ± 0.31
BL (depth=1)	89.52 ± 0.16	66.18 ± 0.20	91.56 ± 0.04	57.06 ± 0.10
E-MLP (depth=2)	89.29 ± 0.20	62.48 ± 0.76	91.32 ± 0.09	57.47 ± 0.21
BL (depth=2)	89.22 ± 0.20	63.68 ± 0.46	91.39 ± 0.06	57.31 ± 0.27
E-MLP (depth=3)	89.37 ± 0.21	66.82 ± 1.01	91.23 ± 0.07	57.36 ± 0.27
BL (depth=3)	88.80 ± 0.18	64.44 ± 0.52	91.13 ± 0.09	57.16 ± 0.48

References

Rishabh Agarwal, Nicholas Frosst, Xuezhou Zhang, Rich Caruana, and Geoffrey E Hinton. Neural additive models: Interpretable machine learning with neural nets. *arXiv preprint arXiv:2004.13912*, 2020.

Rishabh Agarwal, Levi Melnick, Nicholas Frosst, Xuezhou Zhang, Ben Lengerich, Rich Caruana, and Geoffrey E Hinton. Neural additive models: Interpretable machine learning with neural nets. *Advances in neural information processing systems*, 34:4699–4711, 2021.

Ravindra K Ahuja and James B Orlin. Inverse optimization. *Operations research*, 49(5):771–783, 2001.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2623–2631, 2019.

Genevera I Allen, Luqin Gan, and Lili Zheng. Interpretable machine learning for discovery: Statistical challenges and opportunities. *Annual Review of Statistics and Its Application*, 11, 2023.

David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. *Advances in neural information processing systems*, 31, 2018.

Philip W Anderson. More is different: broken symmetry and the nature of the hierarchical structure of science. *Science*, 177(4047):393–396, 1972.

Elaine Angelino, Nicholas Larus-Stone, Daniel Alabi, Margo Seltzer, and Cynthia Rudin. Learning certifiably optimal rule lists for categorical data. *Journal of Machine Learning Research*, 18(234): 1–78, 2018.

Sercan Ö Arik and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 6679–6687, 2021.

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115, 2020.

W Brian Arthur. Complexity and the economy. In *Handbook of Research on Complexity*. Edward Elgar Publishing, 2009.

Anil Aswani, Zuo-Jun Shen, and Auyon Siddiq. Inverse optimization with noisy data. *Operations Research*, 66(3):870–892, 2018.

Susan Athey, Julie Tibshirani, and Stefan Wager. Generalized random forests. 2019.

Santiago R Balseiro, Omar Besbes, and Gabriel Y Weintraub. Dynamic mechanism design with budget-constrained buyers under limited commitment. *Operations Research*, 67(3):711–730, 2019.

Keith Battocchi, Eleanor Dillon, Maggie Hei, Greg Lewis, Paul Oka, Miruna Oprescu, and Vasilis Syrgkanis. Econml: A python package for ml-based heterogeneous treatment effects estimation. *Version 0.16.0*, 2019.

Patrick Bayer, Fernando Ferreira, and Robert McMillan. A unified framework for measuring preferences for schools and neighborhoods. *Journal of political economy*, 115(4):588–638, 2007.

Leonard Bereska and Efstratios Gavves. Mechanistic interpretability for ai safety–a review. *arXiv preprint arXiv:2404.14082*, 2024.

Steven T Berry, James A Levinsohn, and Ariel Pakes. Automobile prices in market equilibrium: Part i and ii, 1993.

Sandra E Black. Do better schools matter? parental valuation of elementary education. *The quarterly journal of economics*, 114(2):577–599, 1999.

Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1721–1730, 2015.

Timothy CY Chan, Rafid Mahmood, and Ian Yihang Zhu. Inverse optimization: Theory and applications. *Operations Research*, 73(2):1046–1074, 2025.

Chun-Hao Chang, Rich Caruana, and Anna Goldenberg. Node-gam: Neural generalized additive model for interpretable deep learning. *arXiv preprint arXiv:2106.01613*, 2021.

Kenneth Y Chay and Michael Greenstone. Does air quality matter? evidence from the housing market. *Journal of political Economy*, 113(2):376–424, 2005.

Huigang Chen, Totte Harinen, Jeong-Yoon Lee, Mike Yung, and Zhenyu Zhao. Causalml: Python package for causal machine learning. *arXiv preprint arXiv:2002.11631*, 2020.

Daniel de Vassimon Manela, Laura Battaglia, and Robin Evans. Marginal causal flows for validation and inference. *Advances in Neural Information Processing Systems*, 37:9920–9949, 2024.

Gerard Debreu. *Theory of value: An axiomatic analysis of economic equilibrium*, volume 17. Yale University Press, 1959.

Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

Jeffrey A Dubin and Daniel L McFadden. An econometric analysis of residential electric appliance holdings and consumption. *Econometrica: Journal of the Econometric Society*, pp. 345–362, 1984.

Ronald Aylmer Fisher. *The genetical theory of natural selection: a complete variorum edition*. Oxford University Press, 1999.

Randy A Freeman and Petar V Kokotovic. Inverse optimality in robust stabilization. *SIAM journal on control and optimization*, 34(4):1365–1391, 1996.

Jacob R Gardner, Geoff Pleiss, David Bindel, Kilian Q Weinberger, and Andrew Gordon Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems*, 2018.

Stephen Gibbons and Stephen Machin. Valuing rail access using transport innovations. *Journal of urban Economics*, 57(1):148–169, 2005.

Josiah Willard Gibbs. *Elementary principles in statistical mechanics: developed with especial reference to the rational foundations of thermodynamics*. C. Scribner's sons, 1902.

Edward L Glaeser and Joseph Gyourko. The impact of building restrictions on housing affordability. *Federal Reserve Bank of New York, Economic Policy Review*, 2002:1–19, 2002.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 297–304. JMLR Workshop and Conference Proceedings, 2010.

S-P Han and Olvi L Mangasarian. Exact penalty functions in nonlinear programming. *Mathematical programming*, 17(1):251–269, 1979.

W Michael Hanemann. Discrete/continuous models of consumer demand. *Econometrica: Journal of the Econometric Society*, pp. 541–561, 1984.

Trevor J Hastie. Generalized additive models. *Statistical models in S*, pp. 249–307, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Jennifer L Hill. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, 2011.

Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4 (2):251–257, 1991.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

Jaehwi Jang, Minjae Song, and Daehyung Park. Inverse constraint learning and generalization by transferable reward decomposition. *IEEE Robotics and Automation Letters*, 9(1):279–286, 2023.

William Jevons. *The theory of political economy*. Springer, 2013.

Leo P Kadanoff. Scaling laws for ising models near t c. *Physics Physique Fizika*, 2(6):263, 1966.

Rudolf Emil Kalman. When is a linear control system optimal? 1964.

George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

Amr Kayid, Nicholas Frosst, and Geoffrey E Hinton. Neural additive models library, 2020.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

Edward H Kennedy. Towards optimal doubly robust estimation of heterogeneous causal effects. *Electronic Journal of Statistics*, 17(2):3008–3049, 2023.

Arezou Keshavarz, Yang Wang, and Stephen Boyd. Imputing a convex objective function. In *2011 IEEE international symposium on intelligent control*, pp. 613–619. IEEE, 2011.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pp. 2668–2677. PMLR, 2018.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International conference on machine learning*, pp. 5338–5348. PMLR, 2020.

Mathias Kraus, Daniel Tschernutter, Sven Weinzierl, and Patrick Zschech. Interpretable generalized additive neural networks. *European Journal of Operational Research*, 317(2):303–316, 2024.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Sören R Künzel, Jasjeet S Sekhon, Peter J Bickel, and Bin Yu. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the national academy of sciences*, 116(10):4156–4165, 2019.

Robert J LaLonde. Evaluating the econometric evaluations of training programs with experimental data. *The American economic review*, pp. 604–620, 1986.

Lev Davidovich Landau and Evgenii Mikhailovich Lifshitz. *Statistical Physics: Volume 5*, volume 5. Elsevier, 2013.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 2002.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, Fujie Huang, et al. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Zachary C Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018.

Guiliang Liu, Sheng Xu, Shicheng Liu, Ashish Gaurav, Sriram Ganapathi Subramanian, and Pascal Poupart. A comprehensive survey on inverse constrained reinforcement learning: Definitions, progress and challenges. *arXiv preprint arXiv:2409.07569*, 2024a.

Shicheng Liu and Minghui Zhu. Meta inverse constrained reinforcement learning: Convergence guarantee and generalization analysis. International Conference on Learning Representations, 2024.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y Hou, and Max Tegmark. Kan: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756*, 2024b.

Lennart Ljung and Torkel Glad. On global identifiability for arbitrary model parametrizations. *automatica*, 30(2):265–276, 1994.

Lars Ljungqvist and Thomas J Sargent. *Recursive macroeconomic theory*. MIT press, 2018.

Luca Longo, Mario Brcic, Federico Cabitza, Jaesik Choi, Roberto Confalonieri, Javier Del Ser, Riccardo Guidotti, Yoichi Hayashi, Francisco Herrera, Andreas Holzinger, et al. Explainable artificial intelligence (xai) 2.0: A manifesto of open challenges and interdisciplinary research directions. *Information Fusion*, 106:102301, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Shehryar Malik, Usman Anwar, Alireza Aghasi, and Ali Ahmed. Inverse constrained reinforcement learning. In *International conference on machine learning*, pp. 7390–7399. PMLR, 2021.

Andreu Mas-Colell, Michael Dennis Whinston, Jerry R Green, et al. *Microeconomic theory*, volume 1. Oxford university press New York, 1995.

Daniel McFadden. Conditional logit analysis of qualitative choice behavior. 1972.

Daniel McFadden. Modelling the choice of residential location. 1977.

Maxime Méloux, Silviu Maniu, François Portet, and Maxime Peyrard. Everything, everywhere, all at once: Is mechanistic interpretability identifiable? *arXiv preprint arXiv:2502.20914*, 2025.

Melkamu Mersha, Khang Lam, Joseph Wood, Ali K Alshami, and Jugal Kalita. Explainable artificial intelligence: A survey of needs, techniques, applications, and future direction. *Neurocomputing*, 599:128111, 2024.

Melanie Mitchell. *Complexity: A guided tour*. Oxford university press, 2009.

Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

Whitney K Newey and Daniel McFadden. Large sample estimation and hypothesis testing. *Handbook of econometrics*, 4:2111–2245, 1994.

Andrew Y Ng, Stuart Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, pp. 2, 2000.

Harsha Nori, Samuel Jenkins, Paul Koch, and Rich Caruana. Interpretml: A unified framework for machine learning interpretability. *arXiv preprint arXiv:1909.09223*, 2019.

Daehyung Park, Michael Noseworthy, Rohan Paul, Subhro Roy, and Nicholas Roy. Inferring task goals and constraints using bayesian nonparametric inverse reinforcement learning. In *Conference on robot learning*, pp. 1005–1014. PMLR, 2020.

Ori Plonsky, Reut Apel, Eyal Ert, Moshe Tennenholtz, David Bourgin, Joshua C Peterson, Daniel Reichman, Thomas L Griffiths, Stuart J Russell, Even C Carter, et al. Predicting human decisions with behavioural theories and machine learning. *Nature Human Behaviour*, pp. 1–14, 2025.

Karl Popper. *The logic of scientific discovery*. Routledge, 2005.

Frank Plumpton Ramsey. A mathematical theory of saving. *The economic journal*, 38(152):543–559, 1928.

Zhi-Yong Ran and Bao-Gang Hu. Parameter identifiability in statistical machine learning: a review. *Neural Computation*, 29(5):1151–1203, 2017.

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, 2016.

Ribana Roscher, Bastian Bohn, Marco F Duarte, and Jochen Garcke. Explainable machine learning for scientific insights and discoveries. *Ieee Access*, 8:42200–42216, 2020.

Sherwin Rosen. Hedonic prices and implicit markets: product differentiation in pure competition. *Journal of political economy*, 82(1):34–55, 1974.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019.

Paul Anthony Samuelson. Foundations of economic analysis. *Science and Society*, 13(1), 1948.

Uri Shalit, Fredrik D Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *International conference on machine learning*, pp. 3076–3085. PMLR, 2017.

Claudia Shi, David Blei, and Victor Veitch. Adapting neural networks for the estimation of treatment effects. *Advances in neural information processing systems*, 32, 2019.

Herbert A Simon. A behavioral model of rational choice. *The quarterly journal of economics*, pp. 99–118, 1955.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Tijmen Tieleman. Training restricted boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th international conference on Machine learning*, pp. 1064–1071, 2008.

Berk Ustun and Cynthia Rudin. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*, 102(3):349–391, 2016.

Aad W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Sewall Wright et al. The roles of mutation, inbreeding, crossbreeding, and selection in evolution. 1932.

Markus Wulfmeier, Peter Ondruska, and Ingmar Posner. Maximum entropy deep inverse reinforcement learning. *arXiv preprint arXiv:1507.04888*, 2015.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Ge Yang and Samuel Schoenholz. Mean field residual networks: On the edge of chaos. *Advances in neural information processing systems*, 30, 2017.

Seungil You, David Ding, Kevin Canini, Jan Pfeifer, and Maya Gupta. Deep lattice networks and partial monotonic functions. *Advances in neural information processing systems*, 30, 2017.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

A Related Work

A.1 Interpretability

Interpretability has become increasingly vital in machine learning (Lipton, 2018; Molnar, 2020), especially for scientific domains (Doshi-Velez & Kim, 2017; Roscher et al., 2020). Ensuring interpretability fosters transparency and reproducibility, and may further provide insights into underlying scientific principles. The ideal form of interpretability is **intrinsic interpretability**, in which a model's structure or parameters are directly understandable to humans. However, intrinsic interpretability is challenging to achieve in some widely used high-capacity models such as deep neural networks (LeCun et al., 2015). This has motivated **post-hoc interpretability** methods (Ribeiro et al., 2016; Lundberg & Lee, 2017), which seek to explain a pre-trained black-box model. While more broadly applicable, such explanations are often considered less suitable for scientific research (Rudin, 2019), as they may compromise stability and faithfulness to the model's decision process.

Performance–Interpretability Trade-off. The limited intrinsic interpretability observed in high-capacity models has long been recognized as a central challenge. This is commonly framed as the *performance–interpretability trade-off* (Rudin, 2019; Arrieta et al., 2020), which posits a tension between predictive performance and intrinsic interpretability. High-performing models such as deep neural networks often lack transparency, whereas intrinsically interpretable models struggle to capture complex nonlinear patterns. Several efforts have sought to mitigate the performance–interpretability trade-off, which can be broadly categorized into four groups. (i) Additive models. Classical GAMs (Hastie, 2017), modern GA2Ms/EBMs (Caruana et al., 2015; Nori et al., 2019), and neural variants such as NAM (Agarwal et al., 2021) and NODE-GAM (Chang et al., 2021) preserve interpretability by decomposing predictions into main effects and low-order interactions. (ii) Concept-based models. Concept Bottleneck Models (Koh et al., 2020), TCAV (Kim et al., 2018), and SENN (Alvarez Melis & Jaakkola, 2018) map inputs into human-interpretable latent concepts and use them as intermediate predictors. (iii) Rule- and score-based systems. SLIM (Ustun & Rudin, 2016) and CORELS (Angelino et al., 2018) generate transparent scoring functions or rule lists with provable optimality guarantees. (iv) Shape-constrained networks. Deep Lattice Networks (You et al., 2017) and related monotonic architectures impose monotonicity and calibration constraints to encode domain priors while retaining flexibility.

Limitations in Scientifically Credible Modeling. The above approaches demonstrate strengths, yet two fundamental limitations restrict their applicability in scientific research. First, most methods are tool-centric modifications of machine learning architectures rather than frameworks grounded in scientific theory (e.g., optimization, dynamical systems, conservation laws). As recent surveys emphasize (Roscher et al., 2020; Karniadakis et al., 2021; Allen et al., 2023; Bereska & Gavves, 2024; Longo et al., 2024; Mersha et al., 2024), genuine scientific insight requires models linked to mechanistic principles, yet many interpretability techniques remain detached from such principles. Second, these approaches are typically non-identifiable (Ran & Hu, 2017; Méloux et al., 2025), meaning that multiple distinct parameterizations can explain the same data. This lack of uniqueness undermines their reliability for recovering ground-truth mechanisms and, in statistical terms, complicates consistency guarantees. As a result, the trained model may fail to converge to the true data-generating process as sample size increases (Newey & McFadden, 1994; Van der Vaart, 2000).

Relation to BL. BL also mitigates the performance–interpretability trade-off. Unlike prior methods, it is principle-driven and scientifically grounded, learning interpretable latent optimization structures directly from data. The framework applies broadly to domains where outcomes arise as solutions to (explicit or latent) optimization problems. It is also identifiable: its smooth and monotone variant, Identifiable Behavior Learning (IBL), guarantees identifiability under mild conditions, ensuring the scientific credibility of its explanations and supporting recovery of the ground-truth model under appropriate conditions.

A.2 Data-Driven Inverse Optimization

Inverse optimization (IO) (Ahuja & Orlin, 2001; Chan et al., 2025) is a core paradigm for learning latent optimization problems from observed data. Traditional IO aims to construct objectives or constraints that exactly rationalize a small set of deterministic decisions. In contrast, data-driven IO (Keshavarz et al., 2011; Aswani et al., 2018) focuses on statistically recovering the underlying problem from large-scale, noisy observational data. Inverse optimal control (IOC) (Kalman, 1964; Freeman & Kokotovic, 1996) extends this paradigm to dynamic settings, seeking to infer sequential decision processes from expert trajectories. Within *machine learning*, inverse reinforcement learning (IRL) (Ng et al., 2000; Wulfmeier et al., 2015) and inverse constrained reinforcement learning (ICRL) (Malik et al., 2021; Liu et al., 2024a) are prominent instances of data-driven IOC: Typically, IRL assumes fixed constraints and learns a reward function, whereas ICRL reverses this role. Both require repeatedly solving for (near-)optimal policies and matching with expert demonstrations—incurring high computational cost. In the *behavioral sciences*, particularly economics, numerous studies can be viewed as instances of the data-driven IO paradigm. Foundational work (McFadden, 1972; Dubin & McFadden, 1984; Hanemann, 1984; Berry et al., 1993) and related studies typically posits theoretically grounded, parametric utility maximization problems (UMPs) and estimates their structural parameters from observed behavior.

Relation to BL. The BL framework also falls under the paradigm of data-driven inverse optimization but differs notably from prior related work in both machine learning and behavioral science. Compared with IRL and ICRL, BL does not rely on matching expert-demonstrated policies with the aim of improving task-specific performance. Instead, it is proposed as a general-purpose, scientifically grounded, and intrinsically interpretable framework that operates via low-cost end-to-end training with a hybrid CE–DSM objective. It jointly learns a utility functions and constraints—a direction that has received little attention in IRL and ICRL (Park et al., 2020; Jang et al., 2023; Liu & Zhu, 2024). Meanwhile, in behavioral science, related work typically formulates distinct utility maximization models under varying assumptions for specific decision contexts, and estimate their parameters accordingly. However, to the best of our knowledge, no existing work proposes a structure-free framework for learning UMPs that generalizes across contexts. BL fills this gap with a structure-free, data-driven approach that does not rely on fixed UMP structures.

A.3 Energy-based Models (EBMs)

Energy-based models (EBMs) (LeCun et al., 2006) are a prominent data-driven IO scheme, rooted in the principle of energy minimization from statistical physics. They learn an energy function $E_\theta(x, y)$ that parameterizes the compatibility between inputs and outputs, inducing a Gibbs distribution $p_\theta(y \mid x) \propto \exp\{-E_\theta(x, y)\}$ that favors outcomes corresponding to low-energy solutions. In practice, this energy function is almost always instantiated by high-capacity neural networks, endowing the learned landscape with strong expressive power but also a black-box nature. Training EBMs typically relies on objectives that circumvent the intractable partition function, with classical approaches including contrastive divergence (Hinton, 2002), persistent contrastive divergence (Tieleman, 2008), and noise-contrastive estimation (Gutmann & Hyvärinen, 2010). A particularly influential line of work is score matching (Hyvärinen & Dayan, 2005) and its denoising variant (DSM) (Vincent, 2011), which have underpinned breakthroughs in score-based generative modeling (Song & Ermon, 2019; 2020) and laid the foundation for modern diffusion methods (Song et al., 2020).

Relation to BL. BL and EBMs exhibit a principled correspondence: BL is grounded in behavioral science and rooted in utility maximization, while EBMs are grounded in statistical physics and based on energy minimization. BL adopts several training techniques common to EBMs, such as Gibbs distribution modeling and denoising score matching (DSM). However, the two frameworks differ substantially in model structure. EBMs primarily focus on generative quality and typically employ black-box neural networks to learn an opaque energy function with little regard for interpretability. In contrast, BL is built on the utility maximization problem (UMP) and its equivalence to penalty formulations, yielding a principled and scientifically grounded framework. Its architecture is composed of intrinsically interpretable blocks, each of which can be explicitly expressed in symbolic form as a UMP—a foundational paradigm in behavioral science and a universal optimization framework. These properties enable BL to jointly achieve high predictive performance, intrinsic interpretability, and identifiability, thereby supporting scientifically credible modeling that extends beyond mere generative capability.

B Scientific Explanation of BL(Deep)

BL(Deep) provides a form of interpretability that is consistent with hierarchical optimization systems. In BL, each layer performs a coarse-graining of the optimization structure implemented by the layer below. An intuitive analogy is a corporate organizational hierarchy: lower-layer managers solve their own local optimization problems, while higher-layer managers aggregate and coordinate the outcomes of many such lower-layer problems to achieve broader organizational objectives. BL(Deep) follows the same principle—higher layers summarize, reorganize, and coordinate the solutions formed at lower layers.

This perspective aligns with many scientific domains characterized by multi-level complexity, including (i) the formation of representative behavioral agents in behavioral sciences, and (ii) renormalization in statistical physics, where fine-scale interactions are compressed into effective coarse-scale potentials.

We describe the explanation procedure below. To build intuition, let us first consider a generic hierarchical optimization system—this may refer to a multi-layer organizational structure composed of individual agents, or a multi-scale physical system composed of interacting particles.

Step 1: Bottom-layer interpretation.

Each bottom-layer block is an optimization problem that directly receives inputs from the environment. These blocks correspond to *micro-level behavioral mechanisms*, such as the decision rules of individual agents performing environment-facing tasks in an organization, or the motion laws governing a single particle in statistical physics. Examining these bottom-layer blocks reveals the fundamental optimization principles followed by all units that directly interact with the environment.

Step 2: Layer-wise coarse-graining and micro-to-macro aggregation.

Blocks in the next layer aggregate the outputs of lower-layer optimization problems through a new optimization step, producing a *coarse-grained behavioral summary*. Each higher-level block represents the effective optimization system that emerges from the interactions among many lower-level units, thereby capturing macro-level regularities distilled from micro-level mechanisms.

This micro-to-macro transition is consistent with many well-established scientific principles, including:

- (i) **Aggregation and coordination**: in hierarchical organizations, the outputs of lower-level agents are aggregated, reallocated, and coordinated by higher-level agents to achieve improved organizational objectives.
- (ii) **Coarse-grained observation**: in hierarchical behavioral systems, individual agents are grouped into categories that share characteristic optimization patterns; in statistical physics, many particles collectively form systems whose coarse-grained behavior is governed by effective potentials induced by microscopic interactions.

Step 3: Bottom-up reconstruction.

A global explanation is obtained by tracing the hierarchy upward, following the model's micro-to-macro abstraction path: raw input features → micro-level optimization blocks → macro-level aggregation and coordination or coarse-grained behavioral constructs → macro-level optimization system.

At each layer, we inspect the characteristics of each block and its associated optimization objective, as well as how these optimization problems evolve across layers. This reveals how each higher layer aggregates, coordinates, or coarse-grains the outputs of the layer below. Together, these observations yield a compact multi-scale interpretation in which BL is understood as a hierarchical optimization system.

C Discussion

In this paper, we propose Behavior Learning (BL). Our key **contributions** are threefold. (i) We propose Behavior Learning, a novel general-purpose machine learning framework grounded in behavioral science, which unifies high predictive performance, intrinsic interpretability, identifiability, and scalability. (ii) For scientific research, BL offers a scientifically grounded and identifiable interpretable ML approach for modeling complex phenomena that defy precise formalization. BL applies broadly to scientific disciplines associated with optimization. (iii) At the paradigm level, BL learns from data the optimization structure of either a single optimization problem or a hierarchical composition of problems through distributional modeling, contributing a new methodology to data-driven inverse optimization.

In what follows, we discuss the **limitations and future directions** of Behavior Learning from the perspectives of theoretical foundations, architecture, and applications.

Scalability of theoretical assumptions. The identifiability-related statistical theorems constitute the core theoretical pillars of IBL, ensuring uniqueness of the interpretability and supporting its scientific credibility. Although these results hold under mild conditions, their behavior in large-scale, highly over-parameterized architectures remains less well understood. This highlights the need for systematic investigations into the robustness, potential failure modes, and empirical boundaries of these guarantees when applied to modern large-scale learning systems.

Choice of basis functions. Polynomial basis functions enhance expressivity while preserving symbolic interpretability in BL (Single). However, high-order polynomials may introduce optimization instability, exacerbate sensitivity to initialization and normalization, and complicate training dynamics. Future work may explore alternative basis families—such as trigonometric, spline-based, or neural basis functions—and develop conditioning or normalization strategies that improve numerical stability without sacrificing interpretability.

Interpretable generative modeling. BL integrates several training techniques from energy-based models while retaining intrinsic interpretability, enabling interpretable generative modeling for vision (e.g., image or video generation) and language (e.g., large language models). Extending BL to explicitly generative architectures in which outputs correspond directly to human-understandable and scientifically meaningful blocks represents a compelling direction. Such extensions could yield generative systems with greater transparency, controllability, and scientific credibility compared to traditional black-box models.

Hybrid architectures for partial interpretability. A promising direction for future work is to develop hybrid architectures that integrate BL with black-box models in a principled way to achieve partial interpretability. Three avenues are particularly worth exploring: (i) Feature-level integration. Black-box neural networks can serve as high-capacity feature extractors, while BL operates on the resulting learned representations to impose structured, optimization-based semantics. (ii) Decision-critical integration. BL blocks may be inserted specifically at high-risk or decision-critical components of the model, substantially reducing the interpretability and reliability risks associated with purely black-box architectures. (iii) Mechanism-level integration. Because BL provides an optimization-driven inductive bias aligned with many real-world mechanisms, selectively applying BL to the parts of the system where such inductive bias is essential may yield models that better capture the underlying ground-truth processes while retaining the flexibility of deep networks, thereby improving generalization performance.

BL for scientific and social-scientific modeling. BL represents data as a composition of optimization problems, closely resonating with modeling paradigms in the natural and social sciences. Its competitive performance, intrinsic interpretability, and statistical rigor position BL as a promising framework for scientific machine learning. Future research may apply BL to domains such as statistical physics, evolutionary biology, computational neuroscience, and climate dynamics, as well as behavioral science, economics, sociology, and political science—particularly in settings involving complex, partially formalized, or cognitively meaningful structures.

D USE OF LARGE LANGUAGE MODELS (LLMS)

We used large language models (LLMs) solely for minor grammar correction and polishing of awkward sentences. They were not used in any other part of the research process.

E Architecture Details

E.1 Learning Scheme Details

Input and output of the BL function. We formulate BL as a direct mapping from input–output pairs to compositional utility representations:

$$\mathrm{BL}: \ \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^{d_{\text{out}}}, \qquad (x, y) \mapsto \mathrm{BL}(x, y) \in \mathbb{R}^{d_{\text{out}}},$$

where the output dimension d_{out} is chosen according to the modeling choice. This formulation intentionally allows BL to return either a scalar or a vector for each (x, y); the following cases are most common:

- Scalar per candidate (pointwise evaluation). Set $d_{\text{out}} = 1$. Here $\mathrm{BL}(x, y) \in \mathbb{R}$ is a scalar compositional utility evaluated for the single candidate y. This view is natural for continuous y (regression or density estimation) or when one prefers to evaluate candidates individually.
- Vectorized over a finite candidate set. If $\mathcal{Y} = \{y_1, \ldots, y_m\}$ is finite, one can choose $d_{\text{out}} = m$ and define the vector-valued output by stacking evaluations over the candidate set:

$$\mathrm{BL}(x) := \begin{bmatrix} \mathrm{BL}(x, y_1) \\ \vdots \\ \mathrm{BL}(x, y_m) \end{bmatrix} \in \mathbb{R}^m.$$

  This vectorized form is convenient for classification: it evaluates all class candidates at once and yields a single compositional utility vector per x.
- Flexibility and equivalence. The scalar and vector modes are compatible: the vectorized form is simply a batch of pointwise evaluations. Conversely, a scalar pointwise evaluator can be used to assemble a vector by repeated calls over a candidate set. The choice between pointwise (scalar) and vectorized outputs is therefore an engineering choice that trades off computational efficiency and convenience.

Given a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, training and inference may use either mode: vectorized computation where feasible (e.g., small finite $\mathcal{Y}$), or pointwise evaluation when $\mathcal{Y}$ is large or continuous.

Conditional Gibbs model. Let $(x, y) \sim \mathcal{D}$ with $x \in \mathbb{R}^d$ and $y = (y^{\text{disc}}, y^{\text{cont}}) \in \mathcal{Y}_{\text{disc}} \times \mathbb{R}^{m_c}$ (discrete, continuous, or hybrid). BL induces a conditional Gibbs distribution with temperature $\tau > 0$:

$$p_\tau(y \mid x) = \frac{\exp\{\mathrm{BL}(x, y)/\tau\}}{Z_\tau(x)}, \qquad Z_\tau(x) = \int_{\mathcal{Y}} \exp\{\mathrm{BL}(x, y')/\tau\}\, dy'.$$

For discrete $\mathcal{Y} = \{y_1, \ldots, y_m\}$, if we choose the vector-output formulation, we define

$$\mathrm{BL}(x) := \big[\mathrm{BL}(x, y_1), \ldots, \mathrm{BL}(x, y_m)\big] \in \mathbb{R}^m,$$

so that the conditional distribution reduces to a softmax over this compositional utility vector:

$$p_\tau(y = k \mid x) = \mathrm{softmax}_k\big(\tfrac{1}{\tau}\, \mathrm{BL}(x)\big)\,.$$

Behaviorally, τ encodes *noisy rationality*; as $\tau \to 0$, $p_\tau(\cdot \mid x)$ concentrates on $\arg\max_y \mathrm{BL}(x, y)$, corresponding to the deterministic optimal choice implied by the learned model.

Supervised, unsupervised, and generative uses. BL accommodates multiple regimes. (i) Supervised: take x as input and y as label. For discrete y, one may either (a) adopt the vector-output formulation, where $\mathrm{BL}(x) \in \mathbb{R}^m$ yields a compositional utility vector over all classes and the likelihood is given by a softmax, or (b) adopt the scalar-output formulation, where $\mathrm{BL}(x, y)$ is evaluated separately for each candidate and then normalized across classes. For continuous y, BL naturally operates in the scalar-output mode, treating $\mathrm{BL}(x, y) \in \mathbb{R}$ as a compositional utility field. (ii) Unsupervised / generative: model a marginal $p(y) \propto \exp\{\mathrm{BL}(y)/\tau\}$ (empty x) or a joint $p(x, y) \propto \exp\{\mathrm{BL}(x, y)/\tau\}$; sampling the Gibbs distribution yields a generator.

Learning objective. Since the response $\mathbf{y}$ may contain both discrete and continuous components, we estimate θ by minimizing a type-specific risk:

$$\mathcal{L}(\theta) = \gamma_{\mathrm{d}}\,\mathbb{E}\big[-\log p_{\mathcal{T}}(y^{\mathrm{disc}} \mid x)\big] \;+\; \gamma_{\mathrm{c}}\,\mathbb{E}\Big\|\nabla_{\tilde{y}^{\mathrm{cont}}} \log p_{\mathcal{T}}(\tilde{y}^{\mathrm{cont}} \mid x) + \sigma^{-2}(\tilde{y}^{\mathrm{cont}} - y^{\mathrm{cont}})\Big\|^2,$$

where the first term is cross-entropy on the discrete component and the second is denoising score matching (DSM) on the continuous component with $\tilde{y}^{\mathrm{cont}} = y^{\mathrm{cont}} + \varepsilon$, $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$. Set $(\gamma_{\mathrm{d}}, \gamma_{\mathrm{c}}) = (1, 0)$ for purely discrete outputs, $(0, 1)$ for purely continuous outputs, and $(> 0, > 0)$ for hybrids.

E.2 Model Structure Details

In the main text we adopted a compact notation for BL; here we present an equivalent, more explicit matrix/vector formulation that makes dimensions, linear maps, and the per-head parameterizations explicit, which is useful for formal proofs and for implementation details.

Fixed bases and head pre-activations. For a block input z (specified below), let

$$m_u(z) \in \mathbb{R}^{d_u}, \qquad m_c(z) \in \mathbb{R}^{d_c}, \qquad m_t(z) \in \mathbb{R}^{d_t}$$

denote fixed basis (e.g., monomial) vectors. Learnable linear maps produce head pre-activations:

$$u(z) := M_u\, m_u(z) + b_u \in \mathbb{R}^{r_u}, \quad c(z) := M_c\, m_c(z) + b_c \in \mathbb{R}^{r_c}, \quad t(z) := M_t\, m_t(z) + b_t \in \mathbb{R}^{r_t},$$

with $M_u \in \mathbb{R}^{r_u \times d_u}$, $M_c \in \mathbb{R}^{r_c \times d_c}$, $M_t \in \mathbb{R}^{r_t \times d_t}$ and optional biases $b_\bullet$.

Single BL block. A single modular block is

$$\mathcal{B}(z) \;=\; \lambda_0^\top\, \phi\big(u(z)\big) \;-\; \lambda_1^\top\, \rho\big(c(z)\big) \;-\; \lambda_2^\top\, \psi\big(t(z)\big), \tag{12}$$

where $\lambda_0 \in \mathbb{R}^{r_u}$, $\lambda_1 \in \mathbb{R}^{r_c}$, $\lambda_2 \in \mathbb{R}^{r_t}$ are learnable weights, and ϕ, ρ, ψ act coordinatewise with the roles specified in Theorem 2.1 (increasing ϕ for utility, penalty ρ for inequality violations, symmetric ψ for equalities). Identifying

$$U_{\theta_U}(x, y) = u\big(z = (x, y)\big), \quad \mathcal{C}_{\theta_C}(x, y) = c\big(z = (x, y)\big), \quad \mathcal{T}_{\theta_T}(x, y) = t\big(z = (x, y)\big),$$

substituting into equation 12 recovers the main-text parameterization in equation 4.

Layer of parallel blocks. A layer $\mathbb{B}_\ell$ stacks d_ℓ parallel copies of equation 12 with (possibly) distinct parameters $\theta_{\ell,i}$:

$$\mathbb{B}_\ell(z_\ell) \;:=\; \begin{bmatrix} \mathcal{B}_{\theta_{\ell,1}}(z_\ell) \\ \vdots \\ \mathcal{B}_{\theta_{\ell,d_\ell}}(z_\ell) \end{bmatrix} \in \mathbb{R}^{d_\ell}.$$

We adopt the standard layered (feedforward) form:

$$z_1 := (x, y), \qquad z_{\ell+1} := \mathbb{B}_\ell(z_\ell) \quad (\ell = 1, \dots, L-1),$$

so that each layer's input is simply the previous layer's output. This is the canonical feedforward architecture.

Optionally, one may allow each layer to explicitly access the original inputs:

$$z_1 := (x, y), \qquad z_{\ell+1} := \mathbb{B}_\ell\big((x, y),\, z_\ell\big).$$

To improve trainability one may also use residual connections:

$$z_{\ell+1} := z_\ell + \mathbb{B}_\ell(z_\ell).$$

Shallow/Deep composition and final affine readout. For depth $L \geq 1$, the BL compositional utility is produced by a final learnable affine transformation of the top layer:

$$\mathrm{BL}(x, y) \;=\; W_L\, \mathbb{B}_L\big(z_L\big) + b_L, \tag{13}$$

with $W_L \in \mathbb{R}^{1 \times d_L}$ for scalar output or $W_L \in \mathbb{R}^{m \times d_L}$ for vector output, and bias b_L of matching dimension. The cases $L = 1$ (with $d_1 = 1$), $L \leq 2$, and $L > 2$ correspond to BL(Single), BL(Shallow), and BL(Deep), respectively, exactly as described in the main text.

E.3 Implementation Details

E.3.1 Function Instantiation

Default instantiation. In practice, we instantiate equation 4 with the specific choice $(\phi, \rho, \psi) = (\tanh, \mathrm{ReLU}, |\cdot|)$:

$$\mathcal{B}(x, y; \theta) = \lambda_0^\top \tanh\big(U_{\theta_U}(x, y)\big) - \lambda_1^\top \mathrm{ReLU}\big(\mathcal{C}_{\theta_C}(x, y)\big) - \lambda_2^\top \big|\mathcal{T}_{\theta_T}(x, y)\big|. \tag{14}$$

Here $\lambda_0, \lambda_1, \lambda_2$ are learnable nonnegative weights. The bounded $\tanh$ captures saturation effects and diminishing returns in the utility head (Jevons, 2013), while ReLU and $|\cdot|$ impose asymmetric (one-sided) and symmetric (two-sided) penalties for inequality and equality violations.

Variants and simplifications. Several variants of equation 14 are often useful:

- Identity utility head. Set $\phi = \mathrm{id}$ so the utility head uses raw polynomials:

$$\mathcal{B} = \lambda_0^\top U_{\theta_U} - \lambda_1^\top \mathrm{ReLU}(\mathcal{C}_{\theta_C}) - \lambda_2^\top |\mathcal{T}_{\theta_T}|.$$

- Smooth penalty alternatives. Replace ReLU with $\mathrm{softplus}$ to yield smooth inequality penalties, or replace $|\cdot|$ with Huber or squared penalties to modulate sensitivity near zero for equality terms.
- Dropping heads. The framework is modular, so one may omit heads depending on the task:
  - No T head: ignores symmetric deviations, yielding a constrained maximization with only inequality penalties.
  - No C head: if the T head is retained, the model reduces to a maximization problem with only equality constraints; if T is also removed, it becomes a fully unconstrained maximization.
  - No U head: produces a pure (soft-)constraint model focusing on feasibility.

  Strikingly, removing both U and T leaves only piecewise-linear ReLU penalties; when followed by a final affine readout, the resulting architecture becomes highly similar to a standard MLP—suggesting that MLPs may be viewed as a closely related special instance within the broader BL framework.

E.3.2 Polynomial Feature Maps and Linear Reductions

We adopt a pragmatic default: use low-degree polynomial maps for single-block models to maximize interpretability, and use affine (degree-1) maps inside blocks for shallow/deep stacks to control parameter growth and compute. Below we state the instantiations and give the final block formulas used in experiments.

BL(Single) — polynomial instantiation. Let $m_D(x, y)$ denote a fixed basis of monomials up to total degree D (e.g. $D \le 2$):

$$m_D(x, y) = \big[x,\ y,\ \mathrm{vec}(xx^\top),\ \mathrm{vec}(xy^\top),\ \mathrm{vec}(yy^\top), \ldots\big]^\top.$$

Parameterize each map as a linear map on this basis:

$$U_{\theta_U}(x, y) = M_U\, m_D(x, y) + b_U, \quad \mathcal{C}_{\theta_C}(x, y) = M_C\, m_D(x, y) + b_C, \quad \mathcal{T}_{\theta_T}(x, y) = M_T\, m_D(x, y) + b_T,$$

with learnable matrices $M_\bullet$ and biases $b_\bullet$. The block becomes

$$\mathcal{B}(x, y; \theta) = \lambda_0^\top \phi(M_U m_D + b_U) - \lambda_1^\top \rho(M_C m_D + b_C) - \lambda_2^\top \psi(M_T m_D + b_T).$$

BL (Shallow/Deep) — linear-by-layer instantiation. For stacked architectures (Shallow/Deep) we use affine maps inside each block to keep per-layer complexity low:

$$U_{\theta_U}(x, y) = A_U\,[x; y] + b_U, \quad \mathcal{C}_{\theta_C}(x, y) = A_C\,[x; y] + b_C, \quad \mathcal{T}_{\theta_T}(x, y) = A_T\,[x; y] + b_T,$$

with learnable $A_\bullet$ and $b_\bullet$. The corresponding block is

$$\mathcal{B}(x, y; \theta) = \lambda_0^\top \phi(A_U[x; y] + b_U) - \lambda_1^\top \rho(A_C[x; y] + b_C) - \lambda_2^\top \psi(A_T[x; y] + b_T).$$

On-demand higher-order terms. If diagnostics or domain knowledge indicate underfitting, we optionally augment the affine maps with selected higher-order terms or interactions. Concretely, this is done by appending a small set of monomials (e.g. $x_i y_j$, x_i^2, y_k^2) to the input vector $[x; y]$ and re-estimating the same affine maps $A_\bullet$. This targeted augmentation preserves the base affine parameterization, increases expressivity only where required, and keeps both computational and statistical costs modest while retaining interpretability.

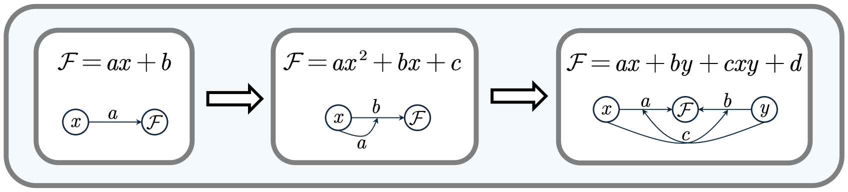

Figure 6: Visualization of polynomial feature maps as computation graphs, where nodes represent variables or outputs and edges represent their effects. The left panel illustrates the linear form $\mathcal{F} = ax + b$, in which the single edge $x \to \mathcal{F}$ directly encodes the marginal effect of x on $\mathcal{F}$. The middle panel shows the quadratic form $\mathcal{F} = ax^2 + bx + c$, where x not only has a direct edge $x \to \mathcal{F}$ but also acts on its own edge ("$x \to \mathcal{F}$"), thereby modifying the strength of its self-effect through a higher-order contribution. The right panel depicts the interaction form $\mathcal{F} = ax + by + cxy + d$, where y has an edge $y \to \mathcal{F}$ and, in addition, x acts on this edge ("$y \to \mathcal{F}$"), thereby modulating the strength of y's contribution to $\mathcal{F}$. Symmetrically, y may act on the edge ("$x \to \mathcal{F}$"), so that each variable can reshape the other's effect through the interaction term.

E.3.3 Skip Connections

Skip connections are optional in our implementation. When beneficial, we often consider two patterns tailored to BL: a DenseNet-style (concatenative) variant and a ResNet-style (additive) variant.

Dense skip connections (DenseNet-style, concatenation). This variant feeds each layer with the concatenation of all preceding representations, mirroring DenseNet (Huang et al., 2017). Let

$$z_1 := [\,x;\, y\,], \qquad s_1 := \mathbb{B}_1(z_1) \in \mathbb{R}^{d_1}.$$

For $\ell \geq 2$,

$$z_\ell := [\,x;\, y;\, s_1; \ldots; s_{\ell-1}\,], \qquad s_\ell := \mathbb{B}_\ell(z_\ell) \in \mathbb{R}^{d_\ell}.$$

The final compositional utility is read out as

$$\mathrm{BL}(x, y) \;=\; W_L\, s_L + b_L.$$

Pros. By exposing all earlier block outputs explicitly as inputs to later blocks, dense skips preserve a transparent feature trail: one can trace which intermediate $\mathcal{B}$-block outputs enter downstream computations and the final affine readout. This often improves feature reuse and yields favorable interpretability at the block level.

Residual skip connections (ResNet-style, addition). This variant adds an identity (or projected) shortcut to each layer, as in ResNet (He et al., 2016). Define

$$z_1 := [\,x;\, y\,], \qquad s_1 := \mathbb{B}_1(z_1) \in \mathbb{R}^{d_1},$$

and for $\ell \geq 2$,

$$s_\ell := \mathbb{B}_\ell(s_{\ell-1}) \;+\; \Pi_\ell\, s_{\ell-1}, \qquad \Pi_\ell \in \mathbb{R}^{d_\ell \times d_{\ell-1}},$$

where Π_ℓ is the identity if $d_\ell = d_{\ell-1}$, or a bias-free learnable projection otherwise. The readout is again

$$\mathrm{BL}(x, y) \;=\; W_L\, s_L + b_L.$$

Skip Connections and Interpretability. Skip connections introduce explicit cross-layer dependency structures, a form widely studied in statistical physics and other scientific domains. Such structures enhance scientific interpretability by making long-range influences transparent. In behavioral and organizational sciences, they capture situations in which lower-level agents directly affect higher-level decision makers without routing through intermediate layers. In physics, microscopic parameters can exert direct effects on macroscopic behaviors across multiple scales. Architecturally, ResNet-style skip connections model linear cross-layer dependencies, whereas DenseNet-style connections realize concatenative (information-replicating) dependencies. These mechanisms provide flexible yet interpretable pathways for representing hierarchical interactions.

F Proofs of Theorems

F.1 Utility Maximization Problem (UMP)

Theorem 2.1 (Penalty Function Equivalence for UMP).

Let $\mathcal{X} \subset \mathbb{R}^{d_x}$ and $\mathcal{Y} \subset \mathbb{R}^{d_y}$ be nonempty compact sets, and let $U : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, $\mathcal{C} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^m$, and $\mathcal{T} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^p$ be Lipschitz continuous. Assume Slater's condition holds for the Utility Maximization Problem (UMP). Then there exist $\lambda_0 > 0$, $\lambda_1 \in \mathbb{R}^m_{++}$, $\lambda_2 \in \mathbb{R}^p_{++}$ such that the unconstrained objective

$$\max_{\mathbf{y}\in\mathcal{Y}} \lambda_0\, \phi\big(U(\mathbf{x},\mathbf{y})\big) - \lambda_1^\top \rho\big(\mathcal{C}(\mathbf{x},\mathbf{y})\big) - \lambda_2^\top \psi\big(\mathcal{T}(\mathbf{x},\mathbf{y})\big) \tag{15}$$

have the same global maximizers. Here, $\phi : \mathbb{R} \to \mathbb{R}$ is a strictly increasing C^1 map, and $\rho, \psi : \mathbb{R} \to \mathbb{R}_{\geq 0}$ are convex "penalty" functions satisfying $\rho(z) = 0$ for $z \leq 0$, $\rho(z) > 0$ for $z > 0$; and $\psi(-z) = \psi(z)$, $\psi(0) = 0$, $\psi(z) > 0$ for $z \neq 0$.

Proof. Let $f(\mathbf{y}) = \phi(U(\mathbf{x},\mathbf{y}))$, $g(\mathbf{y}) = \mathcal{C}(\mathbf{x},\mathbf{y})$, and $h(\mathbf{y}) = \mathcal{T}(\mathbf{x},\mathbf{y})$. By assumption, all functions f, g, h are Lipschitz continuous (since ϕ is C^1 and strictly increasing, its composition with a Lipschitz function remains Lipschitz). Define the feasible set

$$F = \{\mathbf{y} \in \mathcal{Y} \mid g(\mathbf{y}) \leq 0,\ h(\mathbf{y}) = 0\}.$$

By Slater's condition, F is non-empty and has a non-empty interior. The constrained problem (a) is equivalent to:

$$\max_{\mathbf{y}\in\mathcal{Y}} f(\mathbf{y}) \quad \text{s.t.} \quad g(\mathbf{y}) \leq 0,\ h(\mathbf{y}) = 0.$$

Since ϕ is strictly increasing, we have $\arg\max U = \arg\max \phi(U)$. Let $S \subseteq F$ denote the set of global optima of this problem, which is non-empty due to compactness of $\mathcal{Y}$ and continuity of f. Take any $\mathbf{y}^* \in S$, and denote the optimal value by $f^* = f(\mathbf{y}^*)$.

Define the standard penalty function:

$$P_0(\mathbf{y}) = \sum_{i=1}^{m} \rho(g_i(\mathbf{y})) + \sum_{j=1}^{p} \psi(h_j(\mathbf{y})).$$

By properties of ρ and ψ: - $P_0(\mathbf{y}) = 0$ if and only if $\mathbf{y} \in F$, - $P_0(\mathbf{y}) > 0$ if $\mathbf{y} \notin F$.

Under Slater's condition and the Lipschitz continuity of g and h (see Clarke, 1990), there exists a constant $c > 0$ such that:

$$P_0(\mathbf{y}) \geq c \cdot \mathrm{dist}(\mathbf{y}, F), \quad \forall \mathbf{y} \in \mathcal{Y},$$

where $\mathrm{dist}(\mathbf{y}, F) := \inf_{\mathbf{z}\in F} \|\mathbf{y} - \mathbf{z}\|$ denotes the Euclidean distance. This inequality follows from the Mangasarian-Fromovitz constraint qualification (MFCQ), which is implied by Slater's condition.

Let L_f denote the Lipschitz constant of f over $\mathcal{Y}$. For any $\mathbf{y} \in \mathcal{Y}$ and $\mathbf{z} \in F$, we have:

$$f(\mathbf{y}) - f(\mathbf{z}) \leq L_f \|\mathbf{y} - \mathbf{z}\|.$$

Choosing $\mathbf{z} = \mathbf{y}^* \in S$ gives $f(\mathbf{z}) \leq f^*$, so:

$$f(\mathbf{y}) - f^* \leq L_f \|\mathbf{y} - \mathbf{z}\|.$$

Taking infimum over $\mathbf{z} \in F$, we obtain:

$$f(\mathbf{y}) - f^* \leq L_f \cdot \mathrm{dist}(\mathbf{y}, F), \quad \forall \mathbf{y} \in \mathcal{Y}.$$

Combining the inequalities, we get:

$$f(\mathbf{y}) - f^* \leq L_f \cdot \mathrm{dist}(\mathbf{y}, F) \leq \frac{L_f}{c} P_0(\mathbf{y}), \quad \forall \mathbf{y} \in \mathcal{Y}.$$

Let $\mu > \frac{L_f}{c}$. Then for any $\mathbf{y} \notin F$, since $P_0(\mathbf{y}) > 0$, we have:

$$f(\mathbf{y}) - f^* < \mu P_0(\mathbf{y}) \quad \Rightarrow \quad f(\mathbf{y}) - \mu P_0(\mathbf{y}) < f^*.$$

For $\mathbf{y} \in F$, we have $P_0(\mathbf{y}) = 0$ and $f(\mathbf{y}) \leq f^*$, so:

$$f(\mathbf{y}) - \mu P_0(\mathbf{y}) = f(\mathbf{y}) \leq f^*.$$

Hence, the penalized objective $f(\mathbf{y}) - \mu P_0(\mathbf{y})$ satisfies: - $f(\mathbf{y}) - \mu P_0(\mathbf{y}) \leq f^*$ for all $\mathbf{y} \in \mathcal{Y}$, - Equality holds if and only if $\mathbf{y} \in F$ and $f(\mathbf{y}) = f^*$, i.e., $\mathbf{y} \in S$.

Therefore, the unconstrained penalized problem $\max_{\mathbf{y} \in \mathcal{Y}}[f(\mathbf{y}) - \mu P_0(\mathbf{y})]$ shares the same global solution set as the original constrained problem.

Now define the penalty-based score with weights $\lambda_0 = 1$, $\lambda_1 = \mu \mathbf{1}_m$, $\lambda_2 = \mu \mathbf{1}_p$, where $\mathbf{1}_m$ and $\mathbf{1}_p$ denote all-ones vectors of appropriate dimensions. Then:

$$\mathcal{B}(\mathbf{x}, \mathbf{y}) = \lambda_0 \phi(U(\mathbf{x}, \mathbf{y})) - \lambda_1^\top \rho(\mathcal{C}(\mathbf{x}, \mathbf{y})) - \lambda_2^\top \psi(\mathcal{T}(\mathbf{x}, \mathbf{y})) = f(\mathbf{y}) - \mu P_0(\mathbf{y}).$$

Maximizing $\mathcal{B}(\mathbf{x}, \mathbf{y})$ is thus equivalent to maximizing the penalized objective $f(\mathbf{y}) - \mu P_0(\mathbf{y})$, and hence the set of global optima coincides. All weights are positive, as required. □

Theorem 2.2 (Universality of UMP). *Let $\mathcal{X}$ and $\mathcal{Y}$ be arbitrary nonempty sets. Let $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ be an objective and let*

$$\{g_i\}_{i \in I_\leq}, \quad \{\tilde{g}_k\}_{k \in I_\geq}, \quad \{h_j\}_{j \in J}$$

be (possibly empty, countable, or uncountable) families of real–valued constraint functions on $\mathcal{X} \times \mathcal{Y}$. For each fixed $\mathbf{x} \in \mathcal{X}$, consider the optimization problem

$$\sup_{\mathbf{y} \in \mathcal{Y}} f(\mathbf{x}, \mathbf{y}) \quad \text{s.t.} \quad g_i(\mathbf{x}, \mathbf{y}) \leq 0 \ (i \in I_\leq), \ \ \tilde{g}_k(\mathbf{x}, \mathbf{y}) \geq 0 \ (k \in I_\geq), \ \ h_j(\mathbf{x}, \mathbf{y}) = 0 \ (j \in J). \tag{16}$$

Define (with the convention $\sup \varnothing := -\infty$ and maxima taken in the extended reals)

$$U(\mathbf{x}, \mathbf{y}) := f(\mathbf{x}, \mathbf{y}), \qquad \mathcal{C}(\mathbf{x}, \mathbf{y}) := \max\Big\{ 0, \ \sup_{i \in I_\leq} g_i(\mathbf{x}, \mathbf{y}), \ \sup_{k \in I_\geq} \big(-\tilde{g}_k(\mathbf{x}, \mathbf{y})\big) \Big\},$$

$$\mathcal{T}(\mathbf{x}, \mathbf{y}) := \max\Big\{ 0, \ \sup_{j \in J} |h_j(\mathbf{x}, \mathbf{y})| \Big\}.$$

Then for every $\mathbf{x} \in \mathcal{X}$, problem equation 16 is equivalent *to the utility–maximization problem*

$$\sup_{\mathbf{y} \in \mathcal{Y}} U(\mathbf{x}, \mathbf{y}) \quad \text{s.t.} \quad \mathcal{C}(\mathbf{x}, \mathbf{y}) \leq 0, \qquad \mathcal{T}(\mathbf{x}, \mathbf{y}) = 0, \tag{17}$$

in the sense that the feasible sets of equation 16 and equation 17 coincide; hence the optimal values coincide, and whenever maximizers exist, the argmax sets coincide. For minimization problems, replace U by $-f$.

Proof. Fix $\mathbf{x} \in \mathcal{X}$ and write

$$F(\mathbf{x}) := \Big\{ \mathbf{y} \in \mathcal{Y} : \ g_i(\mathbf{x}, \mathbf{y}) \leq 0 \ \forall i, \ \tilde{g}_k(\mathbf{x}, \mathbf{y}) \geq 0 \ \forall k, \ h_j(\mathbf{x}, \mathbf{y}) = 0 \ \forall j \Big\}$$

for the feasible set of equation 16, and

$$\hat{F}(\mathbf{x}) := \Big\{ \mathbf{y} \in \mathcal{Y} : \ \mathcal{C}(\mathbf{x}, \mathbf{y}) \leq 0, \ \mathcal{T}(\mathbf{x}, \mathbf{y}) = 0 \Big\}$$

for the feasible set of equation 17. We show $F(\mathbf{x}) = \hat{F}(\mathbf{x})$.

(i) $F(\mathbf{x}) \subseteq \hat{F}(\mathbf{x})$. Take $\mathbf{y} \in F(\mathbf{x})$. Then $g_i(\mathbf{x},\mathbf{y}) \leq 0$ for all $i \in I_{\leq}$, so $\sup_{i\in I_{\leq}} g_i(\mathbf{x},\mathbf{y}) \leq 0$. Likewise, $\tilde{g}_k(\mathbf{x},\mathbf{y}) \geq 0$ for all $k \in I_{\geq}$ implies $\sup_{k\in I_{\geq}}(-\tilde{g}_k(\mathbf{x},\mathbf{y})) \leq 0$, and $h_j(\mathbf{x},\mathbf{y}) = 0$ for all $j \in J$ implies $\sup_{j\in J}|h_j(\mathbf{x},\mathbf{y})| = 0$. By the definitions of $\mathcal{C}$ and $\mathcal{T}$,

$$\mathcal{C}(\mathbf{x},\mathbf{y}) = \max\{0, \leq 0, \leq 0\} \leq 0, \qquad \mathcal{T}(\mathbf{x},\mathbf{y}) = \max\{0,0\} = 0,$$

hence $\mathbf{y} \in \hat{F}(\mathbf{x})$.

(ii) $\hat{F}(\mathbf{x}) \subseteq F(\mathbf{x})$. Take $\mathbf{y} \in \hat{F}(\mathbf{x})$. From $\mathcal{C}(\mathbf{x},\mathbf{y}) \leq 0$ we obtain

$$\sup_{i\in I_{\leq}} g_i(\mathbf{x},\mathbf{y}) \leq 0 \quad \text{and} \quad \sup_{k\in I_{\geq}}(-\tilde{g}_k(\mathbf{x},\mathbf{y})) \leq 0.$$

By the defining property of the supremum, the first inequality yields $g_i(\mathbf{x},\mathbf{y}) \leq 0$ for all $i \in I_{\leq}$; the second yields $\tilde{g}_k(\mathbf{x},\mathbf{y}) \geq 0$ for all $k \in I_{\geq}$. From $\mathcal{T}(\mathbf{x},\mathbf{y}) = 0$ and $\mathcal{T} \geq 0$ we have $\sup_{j\in J}|h_j(\mathbf{x},\mathbf{y})| \leq 0$, hence $|h_j(\mathbf{x},\mathbf{y})| = 0$ for all $j \in J$, i.e., $h_j(\mathbf{x},\mathbf{y}) = 0$ for all j. Therefore $\mathbf{y} \in F(\mathbf{x})$.

From (i) and (ii) it follows that $F(\mathbf{x}) = \hat{F}(\mathbf{x})$. Since $U = f$ (or $U = -f$ for minimization), the two problems optimize the same objective over the same feasible set; consequently the optimal values agree, and whenever maximizers exist, the argmax sets coincide. □

F.2 BL Architecture

Theorem 2.3 (Universal Approximation of BL). *Let $\mathcal{X} \subset \mathbb{R}^d$ and $\mathcal{Y} \subset \mathbb{R}^m$ be compact sets, and let $p^\star(\mathbf{y} \mid \mathbf{x})$ be any continuous conditional density such that $p^\star(\mathbf{y} \mid \mathbf{x}) > 0$ for all $(\mathbf{x},\mathbf{y}) \in \mathcal{X} \times \mathcal{Y}$. Then for any $\tau > 0$ and $\varepsilon > 0$, there exists a finite BL architecture (with some depth and width depending on ε) and a parameter $\theta^\star$ such that the Gibbs distribution*

$$p_\tau(\mathbf{y} \mid \mathbf{x}; \theta^\star) = \frac{\exp\big(BL_{\theta^\star}(\mathbf{x},\mathbf{y})/\tau\big)}{\int_{\mathcal{Y}} \exp\big(BL_{\theta^\star}(\mathbf{x},\mathbf{y}')/\tau\big)\mathrm{d}\mathbf{y}'} \tag{18}$$

satisfies

$$\sup_{\mathbf{x}\in\mathcal{X}} \mathrm{KL}\big(p^\star(\cdot \mid \mathbf{x}) \,\|\, p_\tau(\cdot \mid \mathbf{x}; \theta^\star)\big) < \varepsilon. \tag{19}$$

Proof. Let $f(\mathbf{x},\mathbf{y}) = \log p^\star(\mathbf{y} \mid \mathbf{x})$. Since $p^\star$ is continuous and strictly positive on the compact domain $\mathcal{X} \times \mathcal{Y}$, the function f is continuous and bounded.

Consider the elementary block $\mathcal{B}_\theta$ in equation 4. Setting $\lambda_1 = \lambda_2 = \mathbf{0}$ and letting U_{θ_U} produce affine features in $(\mathbf{x},\mathbf{y})$, the block reduces to a one-hidden-layer network

$$\mathcal{B}_\theta(\mathbf{x},\mathbf{y}) = \sum_{j=1}^{k} \lambda_{0,j}\, \phi\big(a_j^\top[\mathbf{x};\mathbf{y}] + b_j\big),$$

which is a universal approximator on $C(\mathcal{X} \times \mathcal{Y})$ for nonpolynomial bounded ϕ (Hornik, 1991). Hence, for any $\delta > 0$ there exist θ and a continuous $k(\mathbf{x})$ such that

$$\sup_{(\mathbf{x},\mathbf{y})\in\mathcal{X}\times\mathcal{Y}} \big|\mathcal{B}_\theta(\mathbf{x},\mathbf{y}) - \tau f(\mathbf{x},\mathbf{y}) - k(\mathbf{x})\big| < \delta. \tag{20}$$

Writing $\mathcal{B}_\theta = \tau f + k + \epsilon$ with $|\epsilon| < \delta$, we have

$$\exp\big(\tfrac{\mathcal{B}_\theta}{\tau}\big) = \exp(f)\, \exp(k/\tau)\, \exp(\epsilon/\tau).$$

Let $r = \exp(\epsilon/\tau) - 1$, so $|r| \leq e^{\delta/\tau} - 1 =: B(\delta)$. The normalizer satisfies

$$Z(\mathbf{x}) = \exp(k(\mathbf{x})/\tau)\, \mathbb{E}_{p^\star}[\exp(\epsilon/\tau) \mid \mathbf{x}]\,.$$

With $A(\mathbf{x}) = \mathbb{E}_{p^\star}[r(\mathbf{x},\cdot) \mid \mathbf{x}]$ so that $|A(\mathbf{x})| \leq B(\delta)$, we obtain

$$p_\tau(\mathbf{y} \mid \mathbf{x}; \theta) = p^\star(\mathbf{y} \mid \mathbf{x}) \cdot \frac{1 + r(\mathbf{x},\mathbf{y})}{1 + A(\mathbf{x})}.$$

Define $s(\mathbf{x}, \mathbf{y}) = \frac{1+r}{1+A} - 1$. Then $|s| \le \frac{|r|+|A|}{1-|A|} \le \frac{2B(\delta)}{1-B(\delta)} =: C(\delta)$. If $C(\delta) < \frac{1}{2}$, the inequality $|\log(1+s)| \le 2|s|$ yields

$$\mathrm{KL}(p^\star \,\|\, p_\tau) = -\int p^\star \log(1+s) \le 2\,C(\delta),$$

and the bound is uniform in $\mathbf{x}$. Given $\varepsilon > 0$, pick $\delta > 0$ so that $2C(\delta) < \varepsilon$, proving the claim for Gibbs models with energies from the elementary class.

For the layered architectures, let

$$\mathcal{F}_{\mathcal{B}} := \{\mathcal{B}_\theta : \theta\} \quad \text{and} \quad \mathcal{F}_{\mathrm{BL}} := \{\mathrm{BL}_\theta : \theta\},$$

where BL_θ denotes any finite-depth BL (Shallow $L \le 2$ or Deep $L > 2$) obtained by stacking finitely many parallel $\mathcal{B}_\theta$'s into vectors across layers and applying a final affine map. By construction,

$$\mathcal{F}_{\mathcal{B}} \subseteq \mathcal{F}_{\mathrm{BL}} \quad \text{(take } L = 1 \text{ and the final affine as identity)},$$

hence, under the uniform norm,

$$\overline{\mathcal{F}_{\mathrm{BL}}} \;\supseteq\; \overline{\mathcal{F}_{\mathcal{B}}} \;=\; C(\mathcal{X} \times \mathcal{Y}),$$

and therefore $\overline{\mathcal{F}_{\mathrm{BL}}} = C(\mathcal{X} \times \mathcal{Y})$. The uniform energy approximation argument above then applies verbatim with $g = \mathrm{BL}_{\theta^\star}$, yielding

$$\sup_{\mathbf{x}\in\mathcal{X}} \mathrm{KL}\big(p^\star(\cdot \mid \mathbf{x}) \,\big\|\, p_\tau(\cdot \mid \mathbf{x}; \theta^\star)\big) < \varepsilon.$$

Compactness of $\mathcal{Y}$ and continuity of $\mathrm{BL}_{\theta^\star}$ ensure finiteness of the normalizing constant, so the Gibbs distribution is well defined. This completes the proof. □

F.3 Identifiable Behavior Learning (IBL)

F.3.1 Setup and Assumption

Input–output space and data. Let $\mathcal{X} \subset \mathbb{R}^{d_x}$ and $\mathcal{Y} \subset \mathbb{R}^{d_y}$ be compact sets. Assume the data distribution $P_{X,Y}$ is supported on $\mathcal{X} \times \mathcal{Y}$, and that there exists a point $z_0 = (x_0, y_0)$ in the interior of its support; that is, some open neighborhood of z_0 has positive $P_{X,Y}$-measure. All expectations are taken with respect to $P_{X,Y}$ unless otherwise specified.

Parameter space and polynomial feature maps. The parameter space factorizes as

$$\Theta \;:=\; \Theta_U \times \Theta_C \times \Theta_T \times \mathcal{W}_\circ.$$

For $\theta_U \in \Theta_U$, $\theta_C \in \Theta_C$, and $\theta_T \in \Theta_T$, we define polynomial feature maps

$$p_u : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^{d_u}, \quad p_c : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^{d_c}, \quad p_t : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^{d_t},$$

each of fixed degree and injective in their coefficients (i.e., distinct coefficients yield distinct functions). For a single block, $\theta_U, \theta_C, \theta_T$ correspond to the parameters of the U, C, and T terms together with their respective external multipliers (e.g., penalty weights λ). For a deep network composed of multiple blocks, $\theta = (\theta_U, \theta_C, \theta_T)$ denotes the collection of all block-level parameters across the hierarchy, where θ_U aggregates the parameters of all U-terms, θ_C those of all C-terms, and θ_T those of all T-terms (each including their associated multipliers).

The output component $\mathcal{W}_\circ$ corresponds to the affine transformation in the final layer: $\mathcal{W}_\circ = \mathbb{R}^{d'}$ for single-output prediction, and $\mathcal{W}_\circ = \mathbb{R}^{d' \times m}$ for m-way classification, where d' is the output dimension induced by the preceding network, whether shallow or deep.

Identifiable base block. Let $\lambda_0 \in \mathbb{R}^{d_u}$, $\lambda_1 \in \mathbb{R}^{d_c}$, and $\lambda_2 \in \mathbb{R}^{d_t}$ denote nonnegative weight vectors, treated as learnable parameters. We instantiate the identifiable modular block

$$\mathcal{B}^{\mathrm{id}}(x, y; \theta) \;=\; \lambda_0^\top \tanh\!\big(p_u(x, y)\big) \;-\; \lambda_1^\top \mathrm{softplus}\!\big(p_c(x, y)\big) \;-\; \lambda_2^\top \big(p_t(x, y)\big)^{\odot 2}, \tag{21}$$

where $(\cdot)^{\odot 2}$ denotes elementwise squaring. By construction, the $\tanh$ and $\mathrm{softplus}$ heads are strictly monotone in their arguments, while the quadratic head is even.

We assume that each polynomial feature map $\mathbf{p}_\bullet(x, y)$ contains no nonzero monomial independent of y; that is, no feature is a pure function of x or a constant. This ensures that $\mathcal{B}^{\mathrm{id}}(x, y)$ is nonconstant in y unless all weights vanish.

Architectures. We implement IBL in three architectural forms, each producing a compositional utility function over (x, y).

- IBL(Single): A single block is used as the compositional utility,

$$\mathrm{IBL}(x, y) := \mathcal{B}^{\mathrm{id}}(x, y).$$

- IBL(Shallow): Shallow IBL uses one or two stacked layers of parallel blocks. For instance, a first layer

$$\mathbb{B}_1^{\mathrm{id}}(x, y) := [\,\mathcal{B}_{1,1}^{\mathrm{id}}(x, y), \ldots, \mathcal{B}_{1,d_1}^{\mathrm{id}}(x, y)\,]^\top \in \mathbb{R}^{d_1}$$

  feeds into a bias-free affine map

$$\mathrm{IBL}_{\textsf{Shallow}}(x, y) := \mathbf{W}_1^\circ\, \mathbb{B}_1^{\mathrm{id}}(x, y),$$

  where $\mathbf{W}_1^\circ \in \mathbb{R}^{m \times d_1}$ for classification and $\mathbf{W}_1^\circ \in \mathbb{R}^{1 \times d_1}$ for scalar output.

- IBL(Deep): Deep IBL extends the construction to depth $L > 2$, recursively defined as

$$\mathrm{IBL}(x, y) := \mathbf{W}_L^\circ \cdot \mathbb{B}_L^{\mathrm{id}}\big(\cdots \mathbb{B}_2^{\mathrm{id}}(\mathbb{B}_1^{\mathrm{id}}(x, y)) \cdots \big),$$

  where each $\mathbb{B}_\ell^{\mathrm{id}}$ stacks parallel blocks $\mathcal{B}_{\ell,i}^{\mathrm{id}}(x, y)$, and $\mathbf{W}_L^\circ$ is a bias-free affine transformation. The cases $L = 1$ and $L = 2$ recover the Single and Shallow architectures, respectively.

Induced conditional model. Let $\mathrm{IBL}(x, y)$ denote the compositional utility function produced by the chosen architecture (Single, Shallow, or Deep). It induces the conditional Gibbs distribution

$$\text{(Discrete } y \in [m]) \qquad p(y \mid x) = \mathrm{softmax}_y\{\mathrm{IBL}(x, y)\}, \tag{22}$$

$$\text{(Continuous } y) \qquad p(y \mid x) = \frac{\exp\{\mathrm{IBL}(x, y)/\tau\}}{\int_{\mathcal{Y}} \exp\{\mathrm{IBL}(x, \tilde{y})/\tau\}\, d\tilde{y}}, \quad \tau > 0 \text{ fixed.} \tag{23}$$

Here τ is a fixed temperature parameter. Thus, IBL predicts by defining a compositional utility landscape whose Gibbs distribution governs y given x.

Quotient parameter space.

Definition F.1 (Symmetry Quotient Space)**.** *Define the equivalence relation $\sim$ on Θ as the smallest relation satisfying*

$$\theta_t \sim \theta_t' \quad \Longleftrightarrow \quad p_t^{(i)}(x, y; \theta_t^{(i)})^{\odot 2} = p_t^{(i)}(x, y; \theta_t'^{(i)})^{\odot 2} \quad \textit{for all } i \textit{ and } (x, y).$$

The corresponding quotient space is

$$\bar{\Theta} := \Theta / \sim\,.$$

Explanation. The T-component is designed to encode equality constraints, which are symbolically equations. Flipping the overall sign of such a constraint leaves the equation unchanged, so different parameterizations that differ only by sign should be regarded as equivalent.

Definition F.2 (Scale-Invariant Quotient Space)**.** *Define the equivalence relation $\approx$ on $\bar{\Theta}$ by*

$$\bar{\theta} \approx \bar{\theta}' \quad \Longleftrightarrow \quad \exists\, c > 0 \ \textit{ such that } \ \mathsf{s}(x, y; \bar{\theta}) = c\, \mathsf{s}(x, y; \bar{\theta}').$$

The scale-invariant quotient space is then given by

$$\widetilde{\Theta} := \bar{\Theta} / \approx\,.$$

Explanation. In classification, predictions depend only on relative compositional utility differences between candidate labels. From a technical perspective, quotienting out global shifts or uniform scalings is necessary: without this identification, the cross-entropy loss admits redundant parameterizations that differ only by such transformations. At the same time, this quotient is natural and harmless, since it does not eliminate informative ratios between classes but merely discards absolute levels or scales that play no role in the softmax decision rule.

Loss Functions. We adopt a hybrid loss to simultaneously accommodate discrete and continuous outputs. Specifically, cross-entropy (CE) is applied to discrete targets, while denoising score matching (DSM) is applied to continuous targets. Let $\gamma_c, \gamma_d \geq 0$ with $\gamma_c + \gamma_d > 0$. The population risk, defined on the quotient parameter space, is given by

$$\mathcal{M}(\bar{\theta}) \;=\; \gamma_d \, \mathbb{E}[-\log p_\theta(Y \mid X)] \;+\; \gamma_c \, \mathbb{E}[\mathcal{S}_{\mathrm{DSM}}(\theta; X)] \,, \qquad \theta \in \pi^{-1}(\bar{\theta}), \tag{24}$$

where π denotes the canonical projection from the original parameter space onto its quotient.

For continuous outputs $Y \in \mathcal{Y} \subseteq \mathbb{R}^{d_y}$, DSM is implemented by perturbing the target with additive Gaussian noise $\tilde{Y} = Y + \varepsilon$, $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$, and penalizing the squared discrepancy between the model score and the corresponding denoising score:

$$\mathcal{S}_{\mathrm{DSM}}(\theta; X) = \frac{1}{2\sigma^2} \, \mathbb{E}_\varepsilon \left[\left\| \nabla_{\tilde{y}} \log p_\theta(\tilde{y} \mid X) + \tfrac{1}{\sigma^2}(Y - \tilde{Y}) \right\|^2 \,\middle|\, X, Y \right]. \tag{25}$$

In classification-only settings we set $\gamma_c = 0$ (pure CE), while in regression-only settings we set $\gamma_d = 0$ (pure DSM).

For a single observation $Z = (X, Y)$, we define the *per-sample loss* as

$$\ell(\theta; Z) := \gamma_d \left[-\log p_\theta(Y \mid X) \right] + \gamma_c \, \mathcal{S}_{\mathrm{DSM}}(\theta; X). \tag{26}$$

The empirical criterion then takes the standard M-estimation form

$$\hat{Q}_n(\theta) \;=\; \frac{1}{n} \sum_{i=1}^{n} \ell(\theta; Z_i), \qquad Z_i = (X_i, Y_i). \tag{27}$$

Key Assumptions.

Assumption F.1 (Global Atomic Independence and Injectivity). *Let $\bar{\Psi}$ be the atomic parameter quotient.*

1. *Injectivity on the quotient. The map $\bar{\Psi} \to \mathbb{R}^{\mathcal{X} \times \mathcal{Y}}$, $\bar{\psi} \mapsto g_{\bar{\psi}}$, is injective.*

2. *Linear Independence._ Atomic linear independence. Any finite collection of pairwise distinct atoms $\{g_{\bar{\psi}_i}\}_{i=1}^r$ with $\bar{\psi}_i \in \bar{\Psi}$ is linearly independent in $\mathbb{R}^{\mathcal{X} \times \mathcal{Y}}$.*

3. *Minimality. In all model instances we only consider minimal representations: no duplicate atoms and its corresponding linear coefficient in the mixture is nonzero.*

4. *Canonical ordering. For each model instance, a fixed canonical ordering is imposed on the atom list.*

Explanation. Assumption F.1 treats each identifiable block $\mathcal{B}^{\mathrm{id}}$ as an *atomic* building unit and imposes four structural requirements on representations built from these atoms. Together, these four conditions define a non-ambiguous, non-redundant, and canonical algebra of atoms: after quotienting by the natural symmetries, every model constructed from $\mathcal{B}$-blocks admits a unique minimal representation (up to the prescribed equivalences). This structural regularity is the foundation on which identifiability statements are built: it guarantees that observing the model output (or the objective it optimizes) allows one, in principle, to recover the underlying atomic components and their coefficients in the appropriate quotient sense.

Practical remark. In practice, these conditions can be encouraged or approximately enforced in two complementary ways. First, the design of atomic classes (choice of polynomial bases, interaction terms, and activation heads) can be chosen so that injectivity and linear independence are more plausible by construction. Second, model selection and post-processing (e.g., pruning atoms with near-zero coefficients, enforcing a deterministic tie-breaking rule for ordering) can be applied after training to realize minimality and canonical ordering. These practical measures make the theoretical assumptions operationally meaningful in empirical applications.

F.3.2 Proof of Theorems

Lemma F.1 (Identifiability of Linear Combinations). *Let Z be a set. For each $j = 1, \dots, m$, let Φ_j be a parameter space and define atomic functions*

$$g_\psi := f(\cdot; \phi_j), \qquad \psi = (j, \phi_j) \in \Psi,$$

where $\Psi := \bigsqcup_{j=1}^{m} \Phi_j$ is the disjoint union. Let $\bar{\Psi}$ be the quotient atomic parameter space, and denote its elements by $\bar{\psi} \in \bar{\Psi}$.

Define the quotient parameter space of the model as

$$\bar{\Xi} := \prod_{j=1}^{m} \big((\mathbb{R} \setminus \{0\}) \times \bar{\Psi} \big), \qquad \bar{\xi} = ((a_1, \bar{\psi}_1), \dots, (a_m, \bar{\psi}_m)).$$

The associated linear combination model is

$$S_{\bar{\xi}} := \sum_{j=1}^{m} a_j g_{\bar{\psi}_j}.$$

By virtue of Assumption F.1, the model is identifiable in the quotient parameter space $\bar{\Xi}$: if $S_{\bar{\xi}} \equiv S_{\bar{\xi}'}$ on Z, then $\bar{\xi} = \bar{\xi}'$.

Proof. Suppose $S_{\bar{\xi}} \equiv S_{\bar{\xi}'}$ on Z, i.e.,

$$\sum_{j=1}^{m} a_j \, g_{(j,\phi_j)} - \sum_{j=1}^{m} a'_j \, g_{(j,\phi'_j)} \equiv 0.$$

Let $\mathcal{U}$ be the set of *distinct* atoms in the quotient $\bar{\Psi}$ that appear on either side, and for each $\bar{\psi} \in \mathcal{U}$ let

$$\beta(\bar{\psi}) := \sum_{j:\,[j,\phi_j]=\bar{\psi}} a_j \;-\; \sum_{j':\,[j',\phi'_{j'}]=\bar{\psi}} a'_{j'}$$

be the net coefficient of $g_{\bar{\psi}}$. Then

$$\sum_{\bar{\psi} \in \mathcal{U}} \beta(\bar{\psi}) \, g_{\bar{\psi}} \;\equiv\; 0.$$

By the *linear independence* condition (Assumption F.1:2) of pairwise distinct atoms in $\bar{\Psi}$, we must have $\beta(\bar{\psi}) = 0$ for all $\bar{\psi} \in \mathcal{U}$.

Furthermore, by the *Minimality* requirement (Assumption F.1:3), each $\bar{\psi}$ appears exactly once on each side and with nonzero coefficient. Thus the two sides must contain the exact same list of coefficient–atom pairs $\{(a_j, \bar{\psi}_j)\}_{j=1}^{m}$, and since a canonical ordering is imposed (Assumption F.1:4), it follows that

$$\bar{\xi} = \bar{\xi}'.$$

□

Theorem F.1 (Identifiability of IBL(Single)). *The IBL(Single) architecture uses the atom set*

$$\big\{ \tanh(p_{u,i}),\ \mathrm{softplus}(p_{c,i}),\ (p_{t,i})^2 \,:\ i = 1, \dots, d_u;\ i = 1, \dots, d_c;\ i = 1, \dots, d_t \big\}.$$

Under Assumption F.1, the model is identifiable in the quotient space $\bar{\Theta}$: if $\mathcal{B}^{\mathrm{id}}_\theta \equiv \mathcal{B}^{\mathrm{id}}_{\theta'}$ on $\mathcal{X} \times \mathcal{Y}$, then $\theta = \theta'$ in $\bar{\Theta}$.

Proof. Write

$$\mathcal{B}^{\mathrm{id}}_\theta = \sum_{j=1}^{m} a_j \, f(\cdot; \phi_j), \qquad m := d_u + d_c + d_t,$$

where each $f(\cdot; \phi_j)$ is one of the atoms $\tanh(p_{u,i})$, $\mathrm{softplus}(p_{c,i})$, or $(p_{t,i})^2$, and a_j is the corresponding entry in $(\lambda_0, \lambda_1, \lambda_2)$, with a fixed ordering over all indices.

If $\mathcal{B}^{\mathrm{id}}_\theta \equiv \mathcal{B}^{\mathrm{id}}_{\theta'}$ on $\mathcal{X} \times \mathcal{Y}$, then Lemma F.1 and Assumption F.1 imply that all atoms and coefficients must agree in the quotient atomic space $\bar{\Psi}$. Since the ordering is fixed, this implies $\theta = \theta'$ in $\bar{\Theta}$. □

Theorem F.2 (Identifiability of IBL(Shallow))**.** *The IBL(Shallow) architecture uses the atom set*

$$\{\, \mathcal{B}^{\mathrm{id}}_{\theta_{1,j}}(x,y) \,\}_{j=1}^{d_1},$$

where each $\mathcal{B}^{\mathrm{id}}_{\theta_{1,j}} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is a single-block IBL module parametrized by $\theta_{1,j} \in \Theta_1$. The full parameter is denoted

$$\theta := \big((\theta_{1,1}, \ldots, \theta_{1,d_1}),\ \mathbf{W}_1^{\circ}\big) \in \Theta := (\Theta_1)^{d_1} \times \mathbb{R}^{m \times d_1}.$$

Under Assumption F.1, the mapping $\theta \mapsto \mathrm{IBL}_{\mathit{Shallow}}$ is identifiable in the quotient space $\bar{\Theta}$: if

$$\mathrm{IBL}_{\mathit{Shallow}}(x,y;\theta) \equiv \mathrm{IBL}_{\mathit{Shallow}}(x,y;\theta') \quad \textit{on } \mathcal{X} \times \mathcal{Y},$$

then

$$\theta = \theta' \quad \textit{in } \bar{\Theta}.$$

Proof. Write the k-th output component as a linear combination of atoms:

$$s_\theta^{(k)}(x,y) = \sum_{j=1}^{d_1} w_j^{(k)}\, \mathcal{B}^{\mathrm{id}}_{\theta_{1,j}}(x,y), \qquad k = 1, \ldots, m,$$

where $w_j^{(k)}$ denotes the (k,j)-th entry of $\mathbf{W}_1^{\circ}$.

Suppose two parameter tuples $(\mathbf{W}_1^{\circ}, \{\theta_{1,j}\}_{j=1}^{d_1})$ and $(\mathbf{W}_1^{\circ}{}', \{\theta'_{1,j}\}_{j=1}^{d_1})$ yield identical vector scores on $\mathcal{X} \times \mathcal{Y}$. Then for each k, we have $s_\theta^{(k)} \equiv s_{\theta'}^{(k)}$ on $\mathcal{X} \times \mathcal{Y}$.

Fix any k. Under Assumption F.1, Lemma F.1 ensures that the coefficient–atom pairs $\{(w_j^{(k)}, \mathcal{B}^{\mathrm{id}}_{\theta_{1,j}})\}_{j=1}^{d_1}$ are uniquely determined (up to equivalence in the quotient $\bar{\Theta}$). In particular, for each $j = 1, \ldots, d_1$, we must have

$$w_j^{(k)} = w_j'^{(k)}, \qquad \mathcal{B}^{\mathrm{id}}_{\theta_{1,j}} \equiv \mathcal{B}^{\mathrm{id}}_{\theta'_{1,j}}.$$

Because this holds for all $k = 1, \ldots, m$, it follows that $\mathbf{W}_1^{\circ} = \mathbf{W}_1^{\circ}{}'$ and $\theta_{1,j} = \theta'_{1,j}$ in the quotient parameter space for all j.

Thus $\theta = \theta'$ in $\bar{\Theta}$, establishing full identifiability under fixed ordering. □

Theorem F.3 (Identifiability of IBL(Deep))**.** *Fix integers $L > 2$ and widths $d_1, \ldots, d_{L-1}$. The IBL(Deep) architecture uses the final-layer atom set*

$$\{\, \mathcal{B}^{\mathrm{id}}_{\vartheta_{L,j}}(x,y) \,\}_{j=1}^{d_L} \subset \mathbb{R}^{\mathcal{X} \times \mathcal{Y}},$$

where each $\mathcal{B}^{\mathrm{id}}_{\vartheta_{L,j}} : \mathbb{R}^{d_{L-1}} \to \mathbb{R}$ is a scalar-valued block applied to the output of layer $L-1$. Only the first-layer blocks ($\ell = 1$) are IBL(Single) modules as in Theorem F.1. For architectures with skip connections, the final-layer atoms can be extended to include skipped features (e.g., from earlier layers), which are treated as elements of $\{\, \mathcal{B}^{\mathrm{id}}_{\vartheta_{L,j}}(x,y) \,\}_{j=1}^{d_L}$.

The full parameter is

$$\theta := \big(\{\vartheta_{\ell,j}\}_{\ell=1,j=1}^{L,d_\ell},\ \mathbf{W}_{\mathit{out}}\big) \in \Theta := \prod_{\ell=1}^{L} (\Theta_1)^{d_\ell} \times \mathbb{R}^{m \times d_L}.$$

Under Assumption F.1, the mapping $\theta \mapsto \mathrm{IBL}_{\mathit{Deep}}(x,y;\theta)$ is identifiable in the quotient space $\bar{\Theta}$.

Proof. Under the given architecture, the IBL(Deep) model ultimately takes the form

$$s^{(k)}(x,y) \;=\; \sum_{j=1}^{d_L} w_j^{(k)}\, \mathcal{B}^{\mathrm{id}}_{\vartheta_{L,j}}(x,y), \qquad k = 1, \ldots, m,$$

where each $\mathcal{B}^{\text{id}}_{\vartheta_{L,j}}$ is a scalar-valued function applied to the output of preceding layers. By treating the set $\{\mathcal{B}^{\text{id}}_{\vartheta_{L,j}}(x, y)\}_{j=1}^{d_L}$ as the atom set, we reduce the model to an IBL(Shallow) form:

$$\mathbf{s}(x, y) = \mathbf{W}_{\text{out}}\, \mathbf{B}_L(x, y).$$

Under Assumption F.1, Theorem F.2 applies, implying that the full parameter $\theta = (\{\vartheta_{\ell,j}\}_{\ell,j}, \mathbf{W}_{\text{out}})$ is identifiable in the quotient space $\bar{\Theta}$. □

Theorem 2.4 (Identifiability of IBL). *Under Assumption F.1, the architectures IBL(Single), IBL (Shallow), and IBL(Deep) are all identifiable in the quotient space $\bar{\Theta}$.*

Proof. Immediate from Theorems F.1, F.2, and F.3. □

Theorem 2.5 (Loss Identifiability of IBL). *Let* $\text{IBL}_\theta(x, y)$ *denote an IBL model, and consider the conditional Gibbs distribution*

$$p_\theta(y \mid x) = \frac{\exp\big(\text{IBL}_\theta(x, y)\big)}{\int_{\mathcal{Y}} \exp\big(\text{IBL}_\theta(x, y')\big)\, dy'}.$$

Define the population risk on the symmetry quotient $\bar{\Theta}$ as in equation 24. Assume that the parameter space Θ is compact. Then, under Assumption F.1, the following holds:

(i) If $\gamma_c > 0$, the risk functional $\mathcal{M}$ admits a unique minimizer in $\bar{\Theta}$. Moreover,

$$\mathcal{M}(\bar{\theta}_1) = \mathcal{M}(\bar{\theta}_2) \implies \bar{\theta}_1 = \bar{\theta}_2.$$

(ii) If $\gamma_c = 0$, the risk functional $\mathcal{M}$ admits a unique minimizer in the scale-invariant quotient $\widetilde{\Theta}$. Moreover,

$$\mathcal{M}(\widetilde{\theta}_1) = \mathcal{M}(\widetilde{\theta}_2) \implies \widetilde{\theta}_1 = \widetilde{\theta}_2.$$

Proof. Under Assumption F.1, the IBL architecture is identifiable modulo the symmetry group defined by $\bar{\Theta}$, as established in Theorem 2.4. Let $\theta^\bullet \in \arg\min_{\theta\in\Theta} \mathcal{M}(\theta)$ and set $p^\star(\cdot \mid x) := p_{\theta^\bullet}(\cdot \mid x)$. Since Θ is compact and the loss $\mathcal{M}$ is continuous, a global minimizer exists. We show that it is unique in the stated quotient.

Case $\gamma_c > 0$. At any minimizer we have both $p_\theta(\cdot \mid x) = p^\star(\cdot \mid x)$ and $\nabla_y \log p_\theta(\cdot \mid x) = \nabla_y \log p^\star(\cdot \mid x)$ a.e. Since

$$\nabla_y \log p_\theta(y \mid x) = \nabla_y \text{IBL}_\theta(x, y) - \nabla_y \log Z_\theta(x) = \nabla_y \text{IBL}_\theta(x, y),$$

(the partition function $Z_\theta(x)$ is y-independent), score equality yields $\nabla_y\big(\text{IBL}_\theta - \text{IBL}_{\theta^\bullet}\big)(y; x) = 0$ a.e. IBL contains no y-independent terms. Therefore,

$$\text{IBL}_\theta(x, y) = \text{IBL}_{\theta^\bullet}(x, y) \quad \text{a.e.}$$

By Theorem 2.4 (identifiability in $\bar{\Theta}$), the minimizer is unique in $\bar{\Theta}$; in particular,

$$\mathcal{M}(\bar{\theta}_1) = \mathcal{M}(\bar{\theta}_2) \implies \bar{\theta}_1 = \bar{\theta}_2.$$

Case $\gamma_c = 0$. Here, $\mathcal{M}$ reduces to the cross-entropy risk, which is minimized if and only if $p_\theta(\cdot \mid x) = p^\star(\cdot \mid x)$ almost everywhere.The cross-entropy loss depends on $\text{IBL}_\theta(x, y)$ only through its relative values across y, and is invariant under additive shifts and positive rescalings of the compositional utility. Hence, the loss depends only on the equivalence class $\widetilde{\theta} \in \widetilde{\Theta}\}$. As a result,

$$\mathcal{M}(\widetilde{\theta}_1) = \mathcal{M}(\widetilde{\theta}_2) \implies \widetilde{\theta}_1 = \widetilde{\theta}_2.$$

i.e., the minimizer is unique in $\widetilde{\Theta}$.

Hence, the minimizer is unique in the stated quotient space. This completes the proof. □

Theorem F.4 (Uniform M-estimation consistency (Newey & McFadden, 1994, Theorem 2.1)). *Let $(\mathcal{A}, d)$ be a compact metric space, and let $\widehat{L}_n : \mathcal{A} \to \mathbb{R}$ be a sequence of random objective functions, with population objective $L : \mathcal{A} \to \mathbb{R}$ such that:*

1. *$L(\alpha)$ is uniquely minimized at $\alpha^\star \in \mathcal{A}$;*

2. *$\mathcal{A}$ is compact;*

3. *$L(\alpha)$ is continuous;*

4. *$\widehat{L}_n(\alpha) \xrightarrow{p} L(\alpha)$ uniformly in $\alpha \in \mathcal{A}$.*

Then any sequence $\hat{\alpha}_n \in \arg\min_{\alpha\in\mathcal{A}} \widehat{L}_n(\alpha)$ satisfies $\hat{\alpha}_n \xrightarrow{p} \alpha^\star$.

Theorem F.5 (Consistency of IBL)**.** *Let $\mathcal{M}$ be the population risk defined in equation 24, and let $\mathcal{M}_n$ denote its empirical analogue. Suppose:*

1. *$\{(X_i, Y_i)\}_{i=1}^n$ are i.i.d. samples;*

2. *Θ is compact;*

3. *$\theta \mapsto \mathcal{M}(\theta)$ is continuous, and the loss class admits an integrable envelope such that*

$$\sup_{\theta\in\Theta} \big|\mathcal{M}_n(\theta) - \mathcal{M}(\theta)\big| \xrightarrow{p} 0;$$

Let Ξ denote the relevant quotient space ($\bar{\Theta}$ if $\gamma_c > 0$, $\widetilde{\Theta}$ if $\gamma_c = 0$), and let $\hat{\theta}_n \in \arg\min_{\theta\in\Theta} \mathcal{M}_n(\theta)$ and $\theta^\bullet \in \arg\min_{\theta\in\Theta} \mathcal{M}(\theta)$. Then

$$\hat{\theta}_n \xrightarrow{p} \theta^\bullet \quad \textit{in } \Xi, \qquad \mathcal{M}(\hat{\theta}_n) \xrightarrow{p} \mathcal{M}(\theta^\bullet).$$

If the model is correctly specified (the data law is realized by some $\theta^\star \in \Theta$), then $\theta^\bullet = \theta^\star$ in Ξ, so $\hat{\theta}_n \xrightarrow{p} \theta^\star$.

Proof. Let Ξ denote the relevant quotient space: $\Xi = \bar{\Theta}$ if $\gamma_c > 0$ and $\Xi = \widetilde{\Theta}$ if $\gamma_c = 0$. Let $\pi : \Theta \to \Xi$ be the canonical quotient map. Since Θ is compact and π is continuous and onto, Ξ is compact. By assumption, $\mathcal{M}$ and $\mathcal{M}_n$ are invariant under the corresponding symmetry, hence they factor through π:

$$\widetilde{\mathcal{M}}(\xi) := \mathcal{M}(\theta), \qquad \widetilde{\mathcal{M}}_n(\xi) := \mathcal{M}_n(\theta) \quad (\text{any } \theta \in \pi^{-1}(\xi)).$$

These are well-defined and continuous on Ξ because $\mathcal{M}$ is continuous on Θ. Moreover,

$$\sup_{\xi\in\Xi} \big|\widetilde{\mathcal{M}}_n(\xi) - \widetilde{\mathcal{M}}(\xi)\big| \le \sup_{\theta\in\Theta} \big|\mathcal{M}_n(\theta) - \mathcal{M}(\theta)\big| \xrightarrow{p} 0,$$

so uniform convergence in probability holds on Ξ.

By Loss Identifiability of IBL (Theorem 2.5), $\widetilde{\mathcal{M}}$ has a unique minimizer $\xi^\bullet \in \Xi$. Let $\hat{\xi}_n \in \arg\min_{\xi\in\Xi} \widetilde{\mathcal{M}}_n(\xi)$ (equivalently, choose $\hat{\theta}_n \in \arg\min_{\theta\in\Theta} \mathcal{M}_n(\theta)$ and set $\hat{\xi}_n = \pi(\hat{\theta}_n)$). Then the conditions of Theorem F.4 hold on the compact metric space (Ξ, d), whence

$$\hat{\xi}_n \xrightarrow{p} \xi^\bullet.$$

Since $\widetilde{\mathcal{M}}$ is continuous on Ξ and $\widetilde{\mathcal{M}}(\hat{\xi}_n) = \mathcal{M}(\hat{\theta}_n)$, $\widetilde{\mathcal{M}}(\xi^\bullet) = \mathcal{M}(\theta^\bullet)$ for any representative $\theta^\bullet \in \pi^{-1}(\xi^\bullet)$, we also obtain

$$\mathcal{M}(\hat{\theta}_n) = \widetilde{\mathcal{M}}(\hat{\xi}_n) \xrightarrow{p} \widetilde{\mathcal{M}}(\xi^\bullet) = \mathcal{M}(\theta^\bullet).$$

If the model is correctly specified (there exists $\theta^\star \in \Theta$ inducing the data law), the strict propriety of the CE/DSM terms implies that the unique minimizer in the quotient is the class of $\theta^\star$; hence $\hat{\theta}_n$ converges in probability to $\theta^\star$ in the corresponding quotient space. □

Theorem F.6 (Universal Approximation of IBL)**.** *Let $\mathcal{X} \subset \mathbb{R}^d$ and $\mathcal{Y} \subset \mathbb{R}^m$ be compact sets, and let $p^\star(y \mid x)$ be any continuous conditional density such that $p^\star(y \mid x) > 0$ for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$. Then for any $\tau > 0$ and $\varepsilon > 0$, there exists a finite IBL architecture (with some depth and width depending on ε) and a parameter $\theta^\star$ such that the Gibbs distribution*

$$p_\tau(y \mid x; \theta^\star) = \frac{\exp\big(\mathit{IBL}_{\theta^\star}(x, y)/\tau\big)}{\int_{\mathcal{Y}} \exp\big(\mathit{IBL}_{\theta^\star}(x, y')/\tau\big)\mathrm{d}y'} \tag{28}$$

satisfies

$$\sup_{x\in\mathcal{X}} \mathrm{KL}\big(p^\star(\cdot \mid x) \,\|\, p_\tau(\cdot \mid x;\theta^\star)\big) < \varepsilon. \tag{29}$$

Proof. The argument follows the same construction as in the proof of Theorem 2.3, with only notational modifications due to the IBL parameterization. For brevity, the details are omitted. □

Lemma F.2 (Sieve Approximation Lemma)**.** *Let $\mathcal{C} : \Theta \to [0,\infty)$ be a complexity measure on the parameter space, and let $(c_n)_{n\geq 1}$ be a nondecreasing sequence with $c_n \uparrow \infty$. Define the sieve*

$$\Theta_n := \{\theta \in \Theta : \mathcal{C}(\theta) \leq c_n\},$$

and for a fixed data-generating distribution $p^\dagger$, set

$$\delta_n(p^\dagger) \;:=\; \inf_{\theta\in\Theta_n} \sup_{x\in\mathcal{X}} \mathrm{KL}\big(p^\dagger(\cdot \mid x) \,\|\, p_\theta(\cdot \mid x)\big).$$

Then the following are equivalent:

1. *Sieve universal approximation: For every $\varepsilon > 0$ there exists a constant $C_\varepsilon < \infty$ such that*

$$\inf_{\theta:\,\mathcal{C}(\theta)\leq C_\varepsilon} \sup_{x\in\mathcal{X}} \mathrm{KL}\big(p^\dagger(\cdot \mid x) \,\|\, p_\theta(\cdot \mid x)\big) \;<\; \varepsilon.$$

2. *Vanishing approximation error: $\delta_n(p^\dagger) \downarrow 0$ as $n \to \infty$.*

Moreover, if each Θ_n is compact and $\theta \mapsto \sup_x \mathrm{KL}(p^\dagger \| p_\theta)$ is continuous on Θ_n, then the infimum in $\delta_n(p^\dagger)$ is attained for every n.

Proof. (1) ⇒ (2). Fix $\varepsilon > 0$ and let $C_\varepsilon(p^\dagger)$ be as in (i). Since $c_n \uparrow \infty$, there exists N such that $c_n \geq C_\varepsilon(p^\dagger)$ for all $n \geq N$. Hence $\Theta_n \supseteq \{\theta : \mathcal{C}(\theta) \leq C_\varepsilon(p^\dagger)\}$ for all $n \geq N$, and therefore

$$\delta_n(p^\dagger) \;=\; \inf_{\theta\in\Theta_n} \sup_x \mathrm{KL}(p^\dagger \| p_\theta) \;\leq\; \inf_{\theta:\,\mathcal{C}(\theta)\leq C_\varepsilon(p^\dagger)} \sup_x \mathrm{KL}(p^\dagger \| p_\theta) \;<\; \varepsilon,$$

for all $n \geq N$. Since (δ_n) is nonincreasing in n (because $\Theta_n \uparrow$), it follows that $\delta_n(p^\dagger) \downarrow 0$.

(2) ⇒ (1). Fix $\varepsilon > 0$. By (ii) choose N such that $\delta_N(p^\dagger) < \varepsilon$. Set $C_\varepsilon(p^\dagger) := c_N$. Then

$$\inf_{\theta:\,\mathcal{C}(\theta)\leq C_\varepsilon(p^\dagger)} \sup_x \mathrm{KL}(p^\dagger \| p_\theta) \;\leq\; \inf_{\theta\in\Theta_N} \sup_x \mathrm{KL}(p^\dagger \| p_\theta) \;=\; \delta_N(p^\dagger) \;<\; \varepsilon,$$

which is (i).

The attainment statement follows immediately from compactness of Θ_n and continuity of $\theta \mapsto \sup_x \mathrm{KL}(p^\dagger \| p_\theta)$ on Θ_n. □

Theorem F.7 (Universal Consistency of IBL)**.** *Consider a parameter space Θ for a class of IBL models, and let $\mathcal{C} : \Theta \to [0,\infty)$ be a lower semi-continuous complexity measure (e.g., network depth, width, or parameter norm). Let $(c_n)_{n\geq 1}$ be a nondecreasing sequence with $c_n \uparrow \infty$, and define the sieve*

$$\Theta_n := \{\theta \in \Theta : \mathcal{C}(\theta) \leq c_n\}.$$

Assume:

1. *The map $\theta \mapsto \sup_x \mathrm{KL}(p^\dagger \| p_\theta)$ is continuous on each compact Θ_n.*

2. *The sequence of empirical minimizers $\{\hat\theta_n\}$ is relatively compact in $\bigcup_n \Theta_n$, as ensured by the uniform LLN together with compactness and continuity.*

Then for any admissible data-generating distribution $p^\dagger$ satisfying the regularity assumptions of Theorem F.6, the IBL posterior sequence $\{p_{\hat\theta_n}\}$ satisfies

$$\sup_{x\in\mathcal{X}} \mathrm{KL}\big(p^\dagger(\cdot \mid x) \,\|\, p_{\hat\theta_n}(\cdot \mid x)\big) \xrightarrow{p} 0,$$

i.e. $\{p_{\hat\theta_n}\}$ converges to $p^\dagger$ uniformly in x (in KL).

Proof. Fix an admissible data law $p^\dagger$ (satisfying the regularity of Theorem F.6). For $\theta \in \bigcup_n \Theta_n$ define

$$F(\theta) \;:=\; \sup_{x\in\mathcal{X}} \mathrm{KL}\big(p^\dagger(\cdot \mid x) \,\|\, p_\theta(\cdot \mid x)\big), \qquad \delta_n \;:=\; \inf_{\theta\in\Theta_n} F(\theta).$$

Then Theorem F.6 and Lemma F.2 together imply that $\delta_n \downarrow 0$. By assumption 1, F is continuous on each compact Θ_n.

Let $\hat{\theta}_n \in \arg\min_{\theta\in\Theta_n} \mathcal{M}_n(\theta)$ be any sequence of ERM solutions. We show $F(\hat{\theta}_n) \xrightarrow{p} 0$.

Step 1 (subsequence reduction and precompactness). Take an arbitrary subsequence $(\hat{\theta}_{n_k})_k$. By assumption 2 there exists a further subsequence, still denoted $(\hat{\theta}_{n_k})_k$, and a (possibly k-dependent) index set $N_k \le n_k$ with a parameter limit $\theta_\infty \in \Theta_N$ (for some finite N) such that $\hat{\theta}_{n_k} \to \theta_\infty$ in probability. Passing to a further subsequence if needed, we may assume $N_k \equiv N$.

Step 2 (risk domination against Θ_N-approximants). For each k pick $\theta_k \in \Theta_N$ with $F(\theta_k) \le \delta_N + 1/k$ (attainment follows from compactness and continuity of F on Θ_N). By the ERM property and uniform LLN on Θ_N,

$$\mathcal{M}(\hat{\theta}_{n_k}) \;\le\; \mathcal{M}(\theta_k) + o_p(1) \qquad (k\to\infty).$$

Assume (w.l.o.g.) the CE component is present with a positive weight, so that the population risk decomposes as

$$\mathcal{M}(\theta) \;=\; \mathrm{const} + \gamma_d\, \mathbb{E}_X\big[\mathrm{KL}\big(p^\dagger(\cdot \mid X) \,\|\, p_\theta(\cdot \mid X)\big)\big] \;+\; \gamma_c\, \mathcal{L}^{\mathrm{DSM}}(\theta),$$

with $\gamma_d > 0$ (the DSM-only case is handled analogously by replacing KL with Fisher divergence). Using $\mathbb{E}_X[\mathrm{KL}(\cdot\|\cdot)] \le F(\cdot)$, we obtain

$$\limsup_{k\to\infty} \mathbb{E}_X\Big[\mathrm{KL}\big(p^\dagger(\cdot \mid X) \,\|\, p_{\hat{\theta}_{n_k}}(\cdot \mid X)\big)\Big] \;\le\; \limsup_{k\to\infty} F(\theta_k) \;\le\; \delta_N.$$

Hence, along the subsequence,

$$\mathbb{E}_X\Big[\mathrm{KL}\big(p^\dagger(\cdot \mid X) \,\|\, p_{\hat{\theta}_{n_k}}(\cdot \mid X)\big)\Big] \xrightarrow{p} 0.$$

Step 3 (identification of the subsequential limit). By continuity of the model map $\theta \mapsto p_\theta(\cdot \mid x)$ (from Theorem F.6 regularity) and bounded convergence,

$$\mathbb{E}_X\big[\mathrm{KL}\big(p^\dagger(\cdot \mid X) \,\|\, p_{\theta_\infty}(\cdot \mid X)\big)\big] = 0.$$

Thus $f(x) := \mathrm{KL}\big(p^\dagger(\cdot \mid x) \,\|\, p_{\theta_\infty}(\cdot \mid x)\big)$ equals 0 for P_X-a.e. x. Since f is continuous on compact $\mathcal{X}$ (by the same regularity) and P_X has full support (admissible law), we conclude $f(x) \equiv 0$ on $\mathcal{X}$, i.e.

$$F(\theta_\infty) = \sup_{x\in\mathcal{X}} f(x) = 0.$$

Step 4 (conclude $F(\hat{\theta}_{n_k}) \to 0$ in probability, hence $F(\hat{\theta}_n) \to 0$ in probability). By assumption 1 continuity of F on Θ_N and $\hat{\theta}_{n_k} \to \theta_\infty$ in probability, we have $F(\hat{\theta}_{n_k}) \xrightarrow{p} F(\theta_\infty) = 0$. Since the original subsequence was arbitrary and every subsequence admits a further subsequence with $F(\hat{\theta}_{n_k}) \xrightarrow{p} 0$, the full sequence satisfies $F(\hat{\theta}_n) \xrightarrow{p} 0$.

Therefore,

$$\sup_{x\in\mathcal{X}} \mathrm{KL}\big(p^\dagger(\cdot \mid x) \,\|\, p_{\hat{\theta}_n}(\cdot \mid x)\big) \xrightarrow{p} 0,$$

i.e. $p_{\hat{\theta}_n}(\cdot \mid x) \to p^\dagger(\cdot \mid x)$ uniformly in x in KL. □

Theorem F.8 (Asymptotic normality of extremum estimators (Newey & McFadden, 1994, Theorem 3.1))**.** *Suppose that the estimator $\hat{\theta}_n$ satisfies $\hat{\theta}_n \xrightarrow{p} \theta_0$, and:*

1. *θ_0 lies in the interior of the parameter space Θ;*

2. *the criterion function $\hat{Q}_n(\theta)$ is twice continuously differentiable in a neighborhood $\mathcal{N}$ of θ_0;*

3. *the score satisfies*

$$\sqrt{n}\,\nabla_\theta \hat{Q}_n(\theta_0) \ \xrightarrow{d}\ \mathcal{N}(0,\Sigma);$$

4. *there exists a function $H(\theta)$, continuous at θ_0, such that*

$$\sup_{\theta\in\mathcal{N}} \big\| \nabla^2_\theta \hat{Q}_n(\theta) - H(\theta) \big\| \ \xrightarrow{p}\ 0;$$

5. *the limiting Hessian $H := H(\theta_0)$ is nonsingular.*

Then the estimator is asymptotically normal:

$$\sqrt{n}\,(\hat{\theta}_n - \theta_0) \ \xrightarrow{d}\ \mathcal{N}\big(0,\ H^{-1}\Sigma H^{-1}\big).$$

Theorem F.9 (Asymptotic Normality of IBL)**.** *Consider the IBL family $p_\theta(y \mid x) \propto \exp(\mathrm{IBL}_\theta(x,y))$ with empirical criterion as in equation 27. Assume $(X_i,Y_i)_{i=1}^n$ are i.i.d. from an admissible data law, and that the true parameter θ_0 is an interior point of a locally identifiable chart. For each observation $Z=(X,Y)$, let $\ell(\theta;Z)$ denote the per-sample loss defined in equation 26, so that $\hat{Q}_n(\theta) = \frac{1}{n}\sum_{i=1}^n \ell(\theta;Z_i)$ and $Q(\theta) := \mathbb{E}[\ell(\theta;Z)]$.*

Suppose, in addition:

1. *Score moments. $s(Z) := \nabla_\theta \ell(\theta_0;Z)$ satisfies $\mathbb{E}[s(Z)] = 0$, $\Sigma := Var(s(Z)) < \infty$, and $\frac{1}{\sqrt{n}}\sum_{i=1}^n s(Z_i) \Rightarrow \mathcal{N}(0,\Sigma)$.*

2. *Derivative envelopes. There exists a neighborhood $\mathcal{N}$ of θ_0 and envelopes G_1, G_2 with $\sup_{\theta\in\mathcal{N}} \|\nabla_\theta \ell(\theta;Z)\| \le G_1(Z)$, $\sup_{\theta\in\mathcal{N}} \|\nabla^2_\theta \ell(\theta;Z)\| \le G_2(Z)$, $\mathbb{E}[G_1^2] + \mathbb{E}[G_2] < \infty$.*

3. *Nondegenerate curvature. $H := \nabla^2_\theta Q(\theta_0)$ exists, is continuous at θ_0, and is positive definite, where $Q(\theta) := \mathbb{E}[\hat{Q}_n(\theta)]$.*

Then, under conditions of Theorem 2.7,

$$\sqrt{n}\,(\hat{\theta}_n - \theta_0) \ \Rightarrow\ \mathcal{N}\big(0,\ H^{-1}\Sigma H^{-1}\big).$$

Proof. We verify the hypotheses of Theorem F.8 with $\hat{Q}_n$ as above.

(i) Interior & consistency. By quotient identifiability, fix a local chart in which the population minimizer admits a unique interior representative θ_0. Consistency $\hat{\theta}_n \xrightarrow{p} \theta_0$ follows from uniform M-estimation consistency for IBL (Theorem F.5).

(ii) C^2 criterion. Since IBL_θ is C^2 in θ, the loss $\ell(\theta;Z)$ is twice continuously differentiable in a neighborhood $\mathcal{N}$ of θ_0, and so is $\hat{Q}_n$.

(iii) Score CLT. By *Score moments*,

$$\sqrt{n}\,\nabla_\theta \hat{Q}_n(\theta_0) \ = \ \frac{1}{\sqrt{n}}\sum_{i=1}^n s(Z_i) \ \Rightarrow\ \mathcal{N}(0,\Sigma).$$

(iv) Hessian limit. By *Derivative envelopes* and dominated convergence,

$$\sup_{\theta\in\mathcal{N}} \big\|\nabla^2_\theta \hat{Q}_n(\theta) - \nabla^2_\theta Q(\theta)\big\| \xrightarrow{p} 0,$$

so Assumption 4 of Theorem F.8 holds with $H(\theta) := \nabla^2_\theta Q(\theta)$, continuous at θ_0.

(v) Nonsingularity. By *Nondegenerate curvature*, $H := H(\theta_0)$ is positive definite.

All assumptions of Theorem F.8 are thus verified; consequently, $\sqrt{n}(\hat{\theta}_n - \theta_0) \Rightarrow \mathcal{N}(0, H^{-1}\Sigma H^{-1})$. □

Theorem F.10 (Efficiency of IBL Estimators)**.** *Under the regularity conditions of Theorem F.9, consider the estimating function associated with the per-sample loss equation 26:*

$$\psi_\theta(Z) \ :=\ \nabla_\theta \ell(\theta;Z), \qquad Z=(X,Y).$$

At any population minimizer $\theta^\star$, the moment condition $\mathbb{E}[\psi_{\theta^\star}(Z)] = 0$ holds. Define the sensitivity and variability matrices

$$J \;:=\; \mathbb{E}\big[\nabla_\theta \psi_\theta(Z)\big]\Big|_{\theta=\theta^\star}, \qquad K \;:=\; \mathrm{Var}\big(\psi_{\theta^\star}(Z)\big).$$

Then the asymptotic covariance of $\hat\theta_n$ is given by the Godambe information matrix (sandwich form):

$$\sqrt{n}\,(\hat\theta_n - \theta^\star) \;\Rightarrow\; \mathcal{N}\big(0,\; J^{-1}KJ^{-1}\big).$$

In particular:

1. *CE-only. If $\gamma_c = 0$ (pure cross-entropy) and the model is correctly specified and regular, then $\psi_\theta(Z)$ coincides (up to sign) with the log-likelihood score $s_\theta(Z)$. Hence $J = -I(\theta^\star)$ and $K = I(\theta^\star)$, where $I(\theta^\star)$ denotes the Fisher information matrix. It follows that*

$$\sqrt{n}(\hat\theta_n - \theta^\star) \;\Rightarrow\; \mathcal{N}\big(0,\, I(\theta^\star)^{-1}\big),$$

so the estimator is asymptotically efficient, attaining the Cramér–Rao lower bound.

2. *CE+DSM or DSM-only. Suppose there exists a nonsingular matrix R (constant in a neighborhood of $\theta^\star$) such that*

$$\psi_{\theta^\star}(Z) \;=\; R\, s_{\theta^\star}(Z) \quad a.s.,$$

where $s_\theta(Z) = \nabla_\theta \log p_\theta(Z)$ denotes the parametric score in a local chart. Then $J = RI(\theta^\star)R^\top$ and $K = RI(\theta^\star)R^\top$, so the sandwich covariance again reduces to $I(\theta^\star)^{-1}$. Hence the estimator remains asymptotically efficient.

Proof. The empirical first–order condition is

$$0 \;=\; \frac{1}{n}\sum_{i=1}^n \psi_{\hat\theta_n}(Z_i), \qquad \psi_\theta(Z) := \nabla_\theta \ell(\theta; Z).$$

A mean–value expansion around the population minimizer $\theta^\star$ yields

$$0 \;=\; S_n + G_n(\hat\theta_n - \theta^\star),$$

where

$$S_n := \frac{1}{n}\sum_{i=1}^n \psi_{\theta^\star}(Z_i), \qquad G_n := \frac{1}{n}\sum_{i=1}^n \nabla_\theta \psi_{\tilde\theta}(Z_i),$$

for some intermediate point $\tilde\theta$ lying on the line segment between $\hat\theta_n$ and $\theta^\star$.

Under the regularity conditions of Theorem F.9, we have

$$G_n \xrightarrow{p} J := \mathbb{E}[\nabla_\theta \psi_{\theta^\star}(Z)], \qquad \sqrt{n}\, S_n \;\Rightarrow\; \mathcal{N}(0, K), \quad K := \mathrm{Var}(\psi_{\theta^\star}(Z)).$$

Since J is nonsingular, G_n is invertible with probability tending to one, and hence

$$\sqrt{n}(\hat\theta_n - \theta^\star) = -\, G_n^{-1}\,\sqrt{n}\, S_n \;\Rightarrow\; \mathcal{N}\big(0,\; J^{-1}K(J^{-1})^\top\big).$$

Because here $\psi_\theta = \nabla_\theta \ell(\theta;\cdot)$, the matrix J coincides with the expected Hessian of the loss, which is symmetric. Thus the asymptotic covariance may equivalently be written as $J^{-1}KJ^{-1}$.

(i) CE-only. When $\gamma_c = 0$, the per-sample loss reduces to $\ell(\theta; Z) = -\log p_\theta(Z)$, so that $\psi_\theta(Z) = -s_\theta(Z)$, with $s_\theta(Z) = \nabla_\theta \log p_\theta(Z)$ denoting the likelihood score. Under correct specification and standard likelihood regularity conditions, the information identities hold:

$$\mathbb{E}[s_{\theta^\star}(Z)] = 0, \qquad \mathrm{Var}(s_{\theta^\star}(Z)) = I(\theta^\star), \qquad -\mathbb{E}[\nabla_\theta s_{\theta^\star}(Z)] = I(\theta^\star).$$

Therefore,

$$K = \mathrm{Var}(\psi_{\theta^\star}(Z)) = I(\theta^\star), \qquad J = \mathbb{E}[\nabla_\theta \psi_{\theta^\star}(Z)] = I(\theta^\star),$$

and the asymptotic covariance simplifies to $I(\theta^\star)^{-1}$. Thus the estimator is asymptotically efficient, attaining the Cramér–Rao lower bound (see also Van der Vaart, 2000, Theorem 5.39).

(ii) CE+DSM or DSM-only under score-span. Suppose there exists a nonsingular matrix R (constant in a neighborhood of $\theta^\star$) such that

$$\psi_{\theta^\star}(Z) \;=\; R\, s_{\theta^\star}(Z) \quad \text{a.s.},$$

where $s_\theta(Z)$ is again the parametric score. In this case,

$$K = \mathrm{Var}(\psi_{\theta^\star}(Z)) = R\, I(\theta^\star)\, R^\top, \qquad J = \mathbb{E}[\nabla_\theta \psi_{\theta^\star}(Z)] = -R\, I(\theta^\star).$$

Consequently,

$$J^{-1}K(J^{-1})^\top = \big(-RI(\theta^\star)\big)^{-1}\big(RI(\theta^\star)R^\top\big)\big(-RI(\theta^\star)\big)^{-\top} = I(\theta^\star)^{-1}.$$

Hence the sandwich covariance reduces to the Fisher information bound, and the estimator is asymptotically efficient. This corresponds to the general efficiency condition for minimum-distance or GMM estimators (see Newey & McFadden, 1994, Section 5): the condition $\psi_{\theta^\star} = R\, s_{\theta^\star}$ is equivalent to their moment–span condition $G'W = C\, G'\Omega^{-1}$ (Newey & McFadden, 1994, Equation 5.4), under which the Godambe information collapses to the Fisher bound.

The two claims are thereby established. □

G IBL-BASED MODEL: CAUSAL BEHAVIOR LEARNING (CAUSALBL)

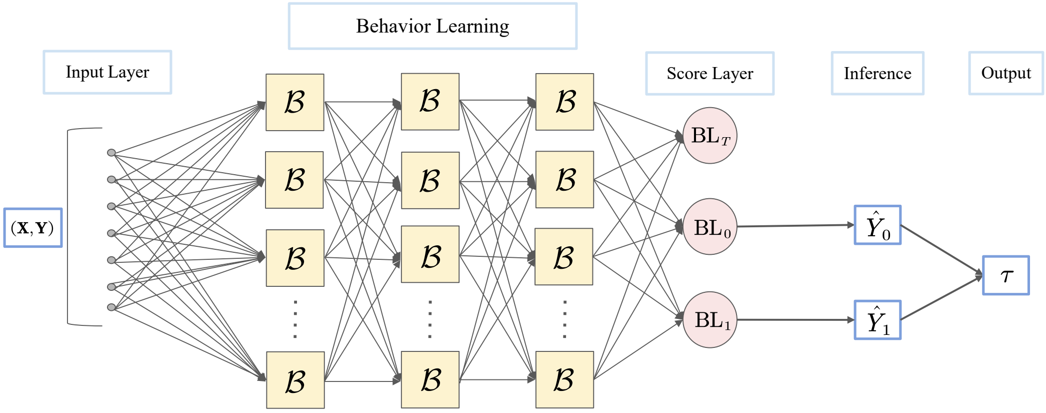

Figure 7: Network architecture of CausalBL. The input layer (x, y) is processed through stacked interpretable blocks $\mathcal{B}$, whose outputs are aggregated in the score layer to form three heads: the treatment head BL_T and the outcome heads $\mathrm{BL}_0, \mathrm{BL}_1$. These heads feed into the inference module to produce the predicted outcomes $\hat{y}_0, \hat{y}_1$ and the treatment effect τ.

Problem setup. Let $x \in \mathcal{X}$ denote covariates, $t \in \{0, 1\}$ a binary treatment (extensions to multi-valued t are straightforward), and y the outcome, which can be discrete with K classes or continuous. For notational simplicity, we write all model-induced distributions as $p(\cdot)$, omitting the dependence on learnable parameters.

CausalBL adopts a *generative* formulation and parameterizes the joint conditional distribution $p(t, y \mid x)$. Within this framework, the quantity $p(t \mid x, y)$ induced by the model is the *posterior factor* of this joint distribution. This posterior factor is used only for parameterizing the joint model and plays no role in causal identification. Following the potential-outcome framework, our causal estimands are the potential outcomes $y_0(x), y_1(x)$ and the resulting *individual treatment effect (ITE)*

$$\tau(x) = y_1(x) - y_0(x).$$

Average treatment effects (ATE) are obtained only by averaging ITE over the test distribution.

Remark G.1. *We emphasize that the posterior factor $p(t \mid x, y)$ is not used for causal identification and never replaces the propensity score; it arises solely from the factorization of the generative model $p(t, y \mid x)$. For continuous outcomes, the dependence of treatment compositional utilities on y stems only from the generative parameterization of the conditional density $p(y \mid x, t)$, not from any causal assumption.*

Design principle. Rather than predicting potential outcomes and treatment assignments with separate discriminative heads, CausalBL parameterizes the factors of the joint conditional distribution $p(t, y \mid x)$ using a collection of *compositional utility heads*. Each head produces a scalar compositional utility, and relative magnitudes are converted into probabilities through a Gibbs/softmax link. Compositional utilities are constructed by stacking interpretable blocks $\mathcal{B}$ (Figure 7), and are grouped into three families:

- Treatment compositional utilities. Two heads $\mathrm{BL}_0^{\mathrm{T}}(\cdot), \mathrm{BL}_1^{\mathrm{T}}(\cdot)$ encode the relative desirability of control vs. treatment. For discrete outcomes, the inputs are only covariates x, i.e. $\mathrm{BL}_j^{\mathrm{T}}(x)$. For continuous outcomes, the treatment utilities also depend on the outcome variable, i.e. $\mathrm{BL}_j^{\mathrm{T}}(x, y)$. The resulting treatment probability is the *posterior factor*

  $$p(t \mid x, y) = \mathrm{softmax}\big(\tfrac{1}{\tau}\,[\mathrm{BL}_0^{\mathrm{T}}(x, y),\, \mathrm{BL}_1^{\mathrm{T}}(x, y)]\big)_t\,,$$

  which is induced by the generative model $p(t, y \mid x)$ and is not a classical propensity score. The predicted treatment is

  $$\hat{t}(x, y) = \arg\max_{j \in \{0,1\}} \mathrm{BL}_j^{\mathrm{T}}(x, y).$$

- Outcome compositional utilities under $t = 0$. For discrete outcomes, we define K heads $\{\mathrm{BL}_k^{(0)}(x)\}_{k=1}^K$, yielding

  $$p(y = k \mid x, t = 0) = \mathrm{softmax}\Big(\tfrac{1}{\tau}\,[\mathrm{BL}_1^{(0)}(x), \ldots, \mathrm{BL}_K^{(0)}(x)]\Big)_k\,, \quad \hat{y}_0(x) = \arg\max_k \mathrm{BL}_k^{(0)}(x)$$

  For continuous outcomes, we instead use a scalar head $\mathrm{BL}^{(0)}(x, y)$ that directly scores candidate outcomes y given x.

- Outcome compositional utilities under $t = 1$. For discrete outcomes, we symmetrically define $\{\mathrm{BL}_k^{(1)}(x)\}_{k=1}^K$. For continuous outcomes, we instead define $\mathrm{BL}^{(1)}(x, y)$, which induces the conditional distribution $p(y \mid x, t = 1)$ and, analogously, the predicted outcome $\hat{y}_1(x)$.

Here $\tau > 0$ is a temperature (fixed or learnable) controlling the sharpness of the softmax/Gibbs link. Grouping compositional utilities by $(\mathrm{T},\ 0,\ 1)$ yields, in the binary case, six heads when $K = 2$ for discrete outcomes (two for t, two for $y \mid t = 0$, two for $y \mid t = 1$); for multi-class y the number of outcome heads scales linearly with K.

Training objectives. Since CausalBL models the joint conditional distribution $p(t, y \mid x)$, the training loss depends on whether the outcome y is discrete or continuous.

Discrete outcomes. When y is discrete, only covariates x are fed into the treatment heads. The posterior factor reduces to $p(t \mid x)$, because $\mathrm{BL}_j^{\mathrm{T}}$ does not depend on y. Training employs a weighted objective based on cross-entropy terms, where probabilities are induced via softmax over compositional utilities.

$$\mathcal{L}_{\mathrm{disc}} = \lambda_{\mathrm{T}}\ \underbrace{\mathrm{CE}\Big(t,\ \mathrm{softmax}\Big(\tfrac{1}{\tau}\,[\mathrm{BL}_0^{\mathrm{T}}(x),\, \mathrm{BL}_1^{\mathrm{T}}(x)]\Big)\Big)}_{\text{posterior factor of the generative model}}$$

$$+\ \lambda_{\mathrm{Y}}\ \Big[\ \mathbb{1}\{t = 0\}\,\mathrm{CE}\Big(y,\ \mathrm{softmax}\Big(\tfrac{1}{\tau}\,\{\mathrm{BL}_k^{(0)}(x)\}_{k=1}^K\Big)\Big) + \mathbb{1}\{t = 1\}\,\mathrm{CE}\Big(y,\ \mathrm{softmax}\Big(\tfrac{1}{\tau}\,\{\mathrm{BL}_k^{(1)}(x)\}_{k=1}^K\Big)\Big)\Big]$$

with nonnegative weights $\lambda_{\mathrm{T}}, \lambda_{\mathrm{Y}}$.

Continuous outcomes. When y is continuous, both x and y are fed into the heads, as the model must parameterize the conditional density $p(y \mid x, t)$. Each branch $t \in \{0, 1\}$ defines a Gibbs-form conditional density

$$p_\tau(y \mid x, t)\ \propto\ \exp\Big(\tfrac{1}{\tau}\,\mathrm{BL}^{(t)}(x, y)\Big)$$

The score $\nabla_y \log p_\tau(y \mid x, t)$ is learned by denoising score matching (DSM). Using Gaussian corruption $\tilde{y} \sim \mathcal{N}(y, \sigma^2 I)$, the DSM loss is

$$\mathcal{L}_{\mathrm{DSM}} = \mathbb{E}\Big[\Big\|\nabla_{\tilde{y}} \log p_\tau(\tilde{y} \mid x, t) + \tfrac{1}{\sigma^2}\,(\tilde{y} - y)\Big\|^2\Big]$$

In addition, treatment posterior factor is trained with cross-entropy:

$$\mathcal{L}_{\text{prop}} = \lambda_{\text{T}}\,\text{CE}\big(t,\ \text{softmax}(\tfrac{1}{\tau}[\text{BL}_0^{\text{T}}(x,y),\ \text{BL}_1^{\text{T}}(x,y)])\big)$$

The total loss for continuous outcomes is

$$\mathcal{L}_{\text{cont}} = \mathcal{L}_{\text{prop}} + \lambda_{\text{DSM}}\,\mathcal{L}_{\text{DSM}}$$

Remark G.2. *This difference from the discrete-outcome case arises purely because, for continuous y, the model must parameterize a density over possible outcomes. Consequently, the treatment utilities $BL_j^{\text{T}}(x,y)$ depend on y only through this generative parameterization of $p(t,y \mid x)$. This dependence does not impose any assumption on the causal structure.*

Mixed outcomes. When y contains both discrete and continuous components, we combine the two objectives by a positive weighted sum:

$$\mathcal{L}_{\text{mixed}} = \gamma_{\text{disc}}\,\mathcal{L}_{\text{disc}} + \gamma_{\text{cont}}\,\mathcal{L}_{\text{cont}}, \quad \gamma_{\text{disc}}, \gamma_{\text{cont}} > 0$$

Amortized inference for fast prediction. For continuous outcomes, or for high-cardinality discrete outcomes, we adopt amortized inference to avoid expensive test-time optimization over y. Importantly, the amortized predictor is trained after CausalBL has been fitted, so that it learns to approximate the optimal solution of the fixed underlying model. Specifically, given covariates x, we learn a branch-specific predictor $g_\psi^{(t)}(x)$ for each treatment t.

The predictor is trained with the following objective:

$$\mathcal{L}_{\text{amort}} = \alpha\,\big[-\text{BL}^{(t)}\big(x,\ g_\psi^{(t)}(x)\big)\big] + \beta\,\|\,y - g_\psi^{(t)}(x)\,\|_2^2, \quad \alpha, \beta \geq 0$$

where the first term encourages the predicted outcome to achieve high compositional utility under the appropriate branch, and the second term ensures numerical accuracy on factual pairs.

Prediction and causal queries.

- Model-induced posterior factor: For discrete outcomes, $\hat{\pi}(x) = \text{softmax}\big(\frac{1}{\tau}[\text{BL}_0^{\text{T}}(x),\ \text{BL}_1^{\text{T}}(x)]\big)_2$. For continuous outcomes, $\hat{\pi}(x,y) = \text{softmax}\big(\frac{1}{\tau}[\text{BL}_0^{\text{T}}(x,y),\ \text{BL}_1^{\text{T}}(x,y)]\big)_2$.
- Potential outcomes: For discrete outcomes, $\hat{y}_t(x) = \arg\max_k \text{BL}_k^{(t)}(x)$, with calibrated class posteriors given by softmax. For continuous outcomes, $\hat{y}_t(x) = g_\psi^{(t)}(x)$, where the amortized predictor is trained after fitting CausalBL.
- Effects: For binary discrete outcomes, report the difference in success probabilities, $\Pr(y = 1 \mid x, t = 1) - \Pr(y = 1 \mid x, t = 0)$. For continuous outcomes, report the individual treatment effect (ITE), $\hat{\tau}(x) = \hat{y}_1(x) - \hat{y}_0(x)$.

Interpretability. Each head is a compositional utility assembled from $\mathcal{B}$-blocks; the value of a head admits a behavioral meaning ("how favorable is class k under branch t"), and the contributions of individual $\mathcal{B}$-blocks can be traced layer-wise. This preserves BL's intrinsic interpretability while enabling causal tasks.

H Experimental Details

H.1 Hardware

Most experiments are conducted on a single NVIDIA L40S GPU. A small number of runs are performed on a laptop equipped with an NVIDIA GeForce RTX 2050 GPU and an Intel Core i7–12700H CPU.

H.2 Standard Prediction Tasks

Datasets. In the Standard Prediction Task, we use 10 OpenML datasets across diverse application domains. Details are given in Table 3.

Table 3: Standard OpenML datasets used in our task. #Features denotes the number of input variables (excluding the target and ID).

Name	Size	#Features	Task type	Field
German Credit	1,000	20	Binary cls.	Finance
Adult Income	48,842	14	Binary cls.	Economics
COMPAS (two-years)	5,278	13	Binary cls.	Law & Society
Bank Marketing	45,211	16	Binary cls.	Marketing
Planning Relax	182	12	Binary cls.	Psychology
EEG Eye State	14,980	14	Binary cls.	Neuroscience
MAGIC Gamma Telescope	19,020	10	Binary cls.	Physics
Electricity	45,312	8	Binary cls.	Electrical Engineering
Wine Quality (Red)	1,599	11	Multiclass	Chemistry
Steel Plates Faults	1,941	27	Multiclass	Industrial Engineering

Baseline Models. For comparison, we include the following baselines: MLP, Neural Additive Model (NAM) (Agarwal et al., 2020; Kayid et al., 2020), ElasticNet, Random Forest, Stochastic Variational Gaussian Process (SVGP) (Gardner et al., 2018), Logistic Regression, Decision Tree, TabNet (Arik & Pfister, 2021), Polynomial Logistic Regression, and LightGBM (Ke et al., 2017).

Table 4: Overview of baseline models in the standard prediction task

Methodological Family	Model Name
Neural networks	Standard MLP Neural Additive Model (NAM) TabNet
Linear regressors	ElasticNet Logistic Regression Polynomial Logistic Regression
Tree-based models	Random Forest Decision Tree
Gradient boosting methods	LightGBM
Bayesian methods	Stochastic Variational Gaussian Process (SVGP)

Data preprocessing. For all ten datasets, we apply a consistent preprocessing strategy. Ordinal categorical variables are mapped to integer levels to preserve their inherent order. Nominal categorical variables without natural ordering are transformed using one-hot encoding. Continuous variables are standardized to zero mean and unit variance. Each dataset is randomly partitioned into train/validation/test splits with a 7:1:2 ratio.

Hyperparameter Tuning Protocol. We perform hyperparameter optimization for most models using the TPE sampler from the Optuna package (Akiba et al., 2019), with 50 trials per dataset. For each model and dataset, the tuned configuration is evaluated under 8 random seeds.

BL Model Hyperparameter Space. For BL(Single) and BL(Shallow), we optimize cross-entropy loss for classification. Both Adam (Kingma, 2014) and AdamW (Loshchilov & Hutter, 2017) optimizers are considered, and the better-performing variant is reported for each dataset. No data augmentation is applied. Batch sizes are chosen in a dataset-specific manner.

- BL(Single): A unified setting is reported across all experiments:

$$\text{degree}_U = [2], \quad \text{degree}_C = [2,2,2], \quad \text{degree}_T = [2,2],$$

$$\sigma_{\text{params}} = 0.01, \quad \sigma_{\lambda_0} = 0.01, \quad \sigma_{\lambda_1} = 0.01, \quad \sigma_{\lambda_2} = 0.01.$$

Here, degree_U, degree_C, and degree_T denote the polynomial degrees of the blocks that parameterize $U(x, y)$, $C(x, y)$, and $T(x, y)$, respectively. Lists indicate both the number of blocks and each block's degree: $\text{degree}_U = [2]$ means a single quadratic block for U, $\text{degree}_C = [2, 2, 2]$ means three quadratic constraint blocks, and $\text{degree}_T = [2, 2]$ means two quadratic belief blocks. σ_{params} initializes coefficients of all polynomial blocks, while σ_{λ_0}, σ_{λ_1}, and σ_{λ_2} initialize the UMP weights $(\lambda_0, \lambda_1, \lambda_2)$. The search grid is reported in Table 5.

- BL(Shallow): We use global gradient clipping of 1.0 and an early stopping patience of 20 epochs without validation improvement. Shallow architectures with depth $L \leq 3$ are considered. The search grid is reported in Table 6.

Baseline Model Hyperparameter Spaces. For baseline models, we also consider both Adam and AdamW for the neural network–based variants, and report results with the better-performing optimizer on each dataset. Batch sizes are tuned separately for each dataset. The detailed hyperparameter search spaces are summarized in Table 7.

Table 5: Hyperparameter tuning space for BL(Single)

Model	Parameter	Search space
BL(Single)	`learning_rate`	{1e−3, 1e−1}
	`batch_size`	{64, 128, 256, 512}
	`max_grad_norm`	{1.0, 2.0, 5.0}

Table 6: Hyperparameter tuning space for BL(Shallow)

Model	Parameter	Search space
BL(Shallow)	`learning_rate`	LogUniform{5e−5, 5e−3}
	`batch_size`	{64, 128, 256, 512}
	`n_layers`	UniformInt{1, 3}
	`n_first_layer`	{24, 30, 36, 40}
	`n_middle_layer`	{8, 6, 4}
	`n_last_layer`	{2, 4, 6}
	`weight_decay`	LogUniform{1e−4, 1e−1}

Table 7: The hyperparameter tuning space for baseline models used in the standard prediction tasks

Model	Parameter	Search space
MLP	`learning_rate`	LogUniform{1e−5, 1e−1}
	`batch_size`	{32, 64, 128, 256}
	`n_layers`	UniformInt{2, 4}
	`hidden_size`	UniformInt{32, 256}
	`weight_decay`	LogUniform{1e−6, 1e−2}
NAM	`learning_rate`	LogUniform{1e−3, 1e−1}
	`batch_size`	{128, 256, 512, 1024}
	`patience`	UniformInt{10, 30}
ElasticNet (SGD)	`alpha`	LogUniform{1e−4, 1e+2}
	`l1_ratio`	Uniform{0.0, 1.0}
	`max_iter`	UniformInt{100, 2000}
	`tol`	LogUniform{1e−6, 1e−2}
	`fit_intercept`	{true, false}
	`learning_rate`	{optimal, constant, invscaling, adaptive}
	`eta0`	LogUniform{1e−4, 1e−1}
	`validation_fraction`	Uniform{0.05, 0.30}
	`n_iter_no_change`	UniformInt{3, 20}

Model	Parameter	Search space
PolyLogistic	degree	{2, 3}
	penalty	{ℓ_2, ℓ_1, ”elasticnet”}
	C	LogUniform{1e−3, 1e+2}
	l1_ratio	Uniform{0.1, 0.9}
	solver	{”liblinear”, ”lbfgs”, “newton-cg”, “saga”}
	max_iter	UniformInt{500, 2000}
	tol	LogUniform{1e−5, 1e−3}
Logistic (ElasticNet)	C	LogUniform{1e−3, 1e+2}
	l1_ratio	Uniform{0.0, 1.0}
	max_iter	UniformInt{100, 2000}
	tol	LogUniform{1e−6, 1e−2}
	fit_intercept	{true, false}
LogisticRegression	solver	{”liblinear”, ”lbfgs”, ”sag”}
	C	LogUniform{1e−4, 1e+2}
	max_iter	UniformInt{100, 2000}
	tol	LogUniform{1e−6, 1e−2}
	fit_intercept	{True, False}
	intercept_scaling	Uniform{0.1, 10.0}
TabNet	learning_rate	LogUniform{1e−4, 3e−2}
	batch_size	{128, 256, 512, 1024}
	virtual_batch_size	{64, 128}
	n_d=n_a	UniformInt{16, 64}
	n_steps	UniformInt{3, 7}
	gamma	Uniform{1.2, 1.7}
	lambda_sparse	LogUniform{1e−6, 1e−3}
DecisionTree	criterion	{”gini”, ”entropy”, ”log_loss”}
	max_depth	UniformInt{3, 20}
	min_samples_split	UniformInt{2, 20}
	min_samples_leaf	UniformInt{1, 10}
	min_weight_fraction_leaf	Uniform{0.0, 0.5}
	max_features	{”sqrt”, ”log2”}
	max_leaf_nodes	UniformInt{10, 1000}
	min_impurity_decrease	Uniform{0.0, 0.1}
	ccp_alpha	Uniform{0.0, 0.1}
GP (SVGP)	kernel	{rbf, matern, rational_quadratic}
	lengthscale	LogUniform{0.1, 10.0}
	rq_alpha	LogUniform{0.1, 5.0}
	num_inducing	UniformInt{100, 500}
	learning_rate	LogUniform{1e−2, 5e−1}
	training_iters	UniformInt{50, 200}
RandomForest	n_estimators	UniformInt{100, 500}
	max_depth	UniformInt{3, 30}
	max_features	{”sqrt”, ”log2”}
	min_samples_leaf	UniformInt{1, 10}
	min_samples_split	UniformInt{2, 20}

H.3 Counterfactual Prediction

H.3.1 Datasets and Synthetic Data Generation.

For the IHDP dataset, we use the benchmark version provided by Shalit et al. (2017), which includes 100 realizations with covariates, treatment assignments, factual and counterfactual outcomes, as well as noiseless potential outcomes that facilitate unbiased evaluation. For the Jobs dataset, we follow the same benchmark format introduced in Shalit et al. (2017), where the variables include covariates, treatment, factual outcomes, and an indicator of randomized assignment.

For the synthetic dataset, we build on the M_3 design in de Vassimon Manela et al. (2024): 10 pretreatment covariates (five gamma, five binary) are generated via a Gaussian-copula dependence with a fixed 10×10 correlation matrix; treatment is binary and the outcome is continuous. On top of this process, we introduce additional nonlinear interactions, heterogeneous treatment effects, and heteroskedastic noise, yielding more complex data-generating processes that better mimic real-world observational studies.

- Nonlinear treatment posterior factor with tunable overlap. We specify

  $$\begin{aligned}\text{logit}\, P(T{=}1 \mid Z) = &-0.3 + 0.1Z_{c1} + 0.2Z_{c2} + 0.5Z_{c3} - 0.2Z_{c4} + Z_{c5} + 0.3Z_{d1} - 0.4Z_{d2} \\ &+ 0.7Z_{d3} - 0.1Z_{d4} + 0.9Z_{d5} + Z_{c2}^2 + 0.5\sin(Z_{c3}) + 0.6Z_{c4}Z_{c5}.\end{aligned}$$

  A temperature parameter multiplies the logit before the sigmoid to control overlap.

- Outcome model with nonlinear functions and heterogeneous treatment effects. Let

  $$g_0(Z) = 0.30Z_{c1} + 0.20Z_{c2}^2 + 0.20\sin(Z_{c3}) - 0.10Z_{c4}Z_{c5},$$

  $$h(Z) = 0.50Z_{c1} - 0.30Z_{c2} + 0.20Z_{c2}^2 + 0.30\sin(Z_{c4}) + 0.20Z_{c1}Z_{c5}.$$

We set $\mu_0(Z) = 1 + g_0(Z)$ and $\tau(Z) = 2 + h(Z)$, so that $\mathbb{E}[Y \mid T, Z] = \mu_0(Z) + T\,\tau(Z)$. Heteroskedastic noise is used: $\sigma(Z) = 0.5 + 0.3\max\{(Z_{c1} + Z_{c2})/2,\, 0\}$ and $Y \sim \mathcal{N}(\mu_0(Z) + T\tau(Z),\, \sigma^2(Z))$.

H.3.2 Baseline Models

For comparison, we evaluate the CausalBL model against several established baselines: Causal Forest (Athey et al., 2019), DragonNet (Shi et al., 2019), TARNet (Shalit et al., 2017), T-Learner, S-Learner, and X-Learner (Künzel et al., 2019), as well as the DR-Learner (Kennedy, 2023).

Table 8: Overview of counterfactual prediction baseline models.

Methodological Family	**Model**
Meta-learners	T-Learner S-Learner X-Learner
Representation-learning networks	DragonNet TARNet
Tree-based models	Causal Forest
Doubly robust estimators	DR-Learner

H.3.3 Metrics

For datasets with known ground-truth treatment effects, we adopt two evaluation metrics: the Rooted Precision in Estimation of Heterogeneous Effect and the Relative ATE error. They are defined as:

$$\sqrt{\epsilon_{\text{PEHE}}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\big(\hat{\tau}(x_i) - \tau(x_i)\big)^2}, \tag{30}$$

$$\epsilon_{\text{relATE}} = \left|\frac{\hat{\tau}_{\text{ATE}} - \tau_{\text{ATE}}}{\tau_{\text{ATE}}}\right|, \tag{31}$$

where $\hat{\tau}(x_i)$ and $\tau(x_i)$ denote the estimated and true individual treatment effects; $\hat{\tau}_{\text{ATE}}$ and τ_{ATE} denote the estimated and true average treatment effects.

For the Jobs dataset, since the ground-truth counterfactual outcomes are unavailable, we employ the Policy Risk $R_{\text{pol}}(\pi)$ (Künzel et al., 2019; Shalit et al., 2017) and the Absolute Treatment Effect on the Treated error (ATT error) (Shalit et al., 2017). Specifically,

$$\begin{aligned} R_{\text{pol}}(\pi_{\hat{\tau}}) = 1 - \Big\{ & p(\pi_{\hat{\tau}}(x) = 1) \cdot \mathbb{E}[Y^1 \mid \pi_{\hat{\tau}}(x) = 1] \\ & + p(\pi_{\hat{\tau}}(x) = 0) \cdot \mathbb{E}[Y^0 \mid \pi_{\hat{\tau}}(x) = 0] \Big\}, \end{aligned} \tag{32}$$

$$\epsilon_{\text{ATT}} = \left| \tau_{\text{ATT}} - \frac{1}{|T|} \sum_{i \in T} \big(f(x_i, 1) - f(x_i, 0) \big) \right|, \tag{33}$$

where τ_{ATT} is the true ATT from the RCT subset, and the second term is the average estimated ITE on the treated units.

H.3.4 Model Hyperparameter Configurations

We summarize the hyperparameter configurations for both CausalBL and the baseline models used in the counterfactual prediction experiments. In general, we adopt the default hyperparameters provided by the respective implementations, with minimal tuning as detailed below.

- CausalBL. Unless otherwise specified, we use a single hidden layer with $h = 12$ units, weight decay $\lambda = 10^{-4}$, and batch size $B = 64$. We apply $n_{\text{noise}} = 1$ denoising score matching (DSM) perturbation per sample with variance σ_{DSM}. Training proceeds in two phases. Phase 1 runs for $E_1 = 100$ epochs with early stopping (patience $p_1 = 10$, improvement threshold $\delta_{\min} = 10^{-4}$). The loss is weighted by λ_{T} (treatment posterior factor loss term) and λ_{Y} (outcome loss term). Phase 2 uses $E_2 = 50$ epochs with early stopping (patience $p_2 = 6$, improvement threshold $\delta_{\min} = 10^{-5}$). The learning rate is fixed at $\eta = 10^{-3}$ for both phases. For model selection, we perform a discrete grid search over

$$\sigma_{\text{DSM}} \in \{1.0, 1.2, 1.5\}, \quad \lambda_{\text{T}} \in \{0.1, 0.2\}, \quad E_1 \in \{50, 100\}, \quad \lambda_{\text{Y}} \in \{0.8, 0.9\}$$

- EconML implementations. For DR-Learner, T-Learner, X-Learner and Causal Forest, we use the implementations provided by the `EconML` package (Battocchi et al., 2019), with the recommended default hyperparameters from the official documentation.

- CausalML implementations. For DragonNet and S-Learner, we adopt the implementations from the `CausalML` package (Chen et al., 2020), using the recommended hyperparameters as in the official Jupyter notebook examples.

- TARNet. We start from the default configuration of Shalit et al. (2017), but apply minor tuning for consistency with DragonNet. Concretely, we use three hidden layers with 200 units in the shared representation and two hidden layers with 100 units in each outcome head, with ELU activations. We also set the batch size to 64, in line with the DragonNet implementation.

H.3.5 Results

IHDP results. Figure 8 reports the forest plots of $\sqrt{\text{PEHE}}$ on the IHDP dataset, with the left and right panels showing within-sample and out-of-sample performance, respectively. Each marker indicates the mean over runs, and the horizontal bar denotes its uncertainty interval. Across both splits, CausalBL consistently ranks among the top methods, outperforming most baselines and achieving performance comparable to the best-performing baseline, T-Learner. In general, CausalBL attains consistently low $\sqrt{\text{PEHE}}$ with narrow intervals, indicating a stable estimation in both settings.

Jobs results. Figure 9 presents box plots of the absolute ATT error on the Jobs dataset, with the left and right panels showing within-sample and out-of-sample performance, respectively. CausalBL is among the top performers, with low median error and small variance, and remains competitive with the strongest baselines across both settings.

Table 9: Within- and out-of-sample performance on the IHDP and Jobs datasets. Reported are the mean ± standard error of $\sqrt{\text{PEHE}}$, ATE relative error, policy risk, and $|\text{ATT}|$ error. Top two results per column are highlighted in blue and red.

Model	Within-sample				Out-of-sample			
	IHDP		Jobs		IHDP		Jobs	
	$\sqrt{\epsilon_{\text{PEHE}}}$	ϵ_{ATE}	R_{POL}	ϵ_{ATT}	$\sqrt{\epsilon_{\text{PEHE}}}$	ϵ_{ATE}	R_{POL}	ϵ_{ATT}
IBL-based Model	1.07 ± 0.62	0.03 ± 0.02	0.23 ± 0.01	0.01 ± 0.01	1.19 ± 0.67	0.05 ± 0.04	0.26 ± 0.03	0.07 ± 0.07
DR-Learner	2.87 ± 1.57	0.10 ± 0.09	0.14 ± 0.02	0.05 ± 0.02	2.65 ± 1.53	0.11 ± 0.10	0.23 ± 0.04	0.08 ± 0.07
DragonNet	1.20 ± 0.62	0.07 ± 0.06	0.23 ± 0.01	0.24 ± 0.14	1.25 ± 0.67	0.07 ± 0.06	0.24 ± 0.04	0.24 ± 0.16
GRF	1.47 ± 1.04	0.05 ± 0.05	0.14 ± 0.01	0.01 ± 0.01	1.50 ± 1.05	0.07 ± 0.08	0.22 ± 0.04	0.08 ± 0.07
S-Learner	1.16 ± 0.87	0.03 ± 0.02	0.17 ± 0.01	0.02 ± 0.01	1.19 ± 0.90	0.05 ± 0.07	0.22 ± 0.04	0.08 ± 0.07
TARNet	1.37 ± 0.28	0.10 ± 0.07	0.20 ± 0.01	0.03 ± 0.02	1.42 ± 0.37	0.11 ± 0.09	0.24 ± 0.04	0.08 ± 0.08
T-Learner	0.88 ± 0.24	0.02 ± 0.02	0.11 ± 0.01	0.01 ± 0.01	0.99 ± 0.39	0.03 ± 0.03	0.22 ± 0.05	0.08 ± 0.06
X-Learner	1.32 ± 0.80	0.03 ± 0.03	0.12 ± 0.01	0.01 ± 0.01	1.36 ± 0.85	0.05 ± 0.05	0.21 ± 0.05	0.08 ± 0.06

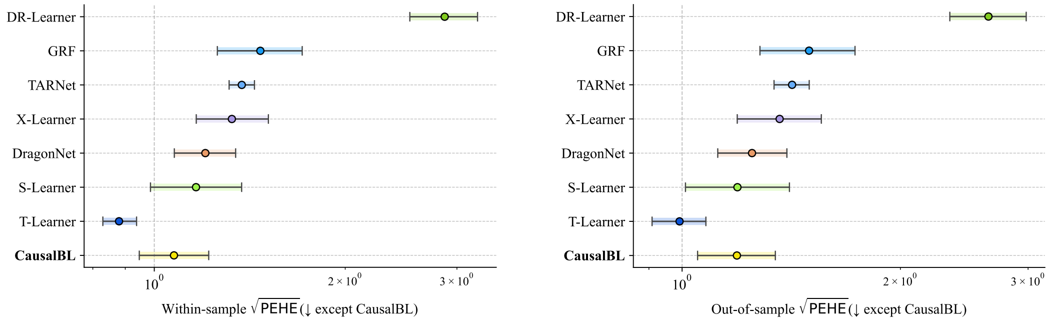

Figure 8: Forest plots of $\sqrt{\text{PEHE}}$ for within-sample (left) and out-of-sample (right) evaluation on the IHDP dataset. Methods are ordered by mean error, with CausalBL fixed at the bottom for clarity.

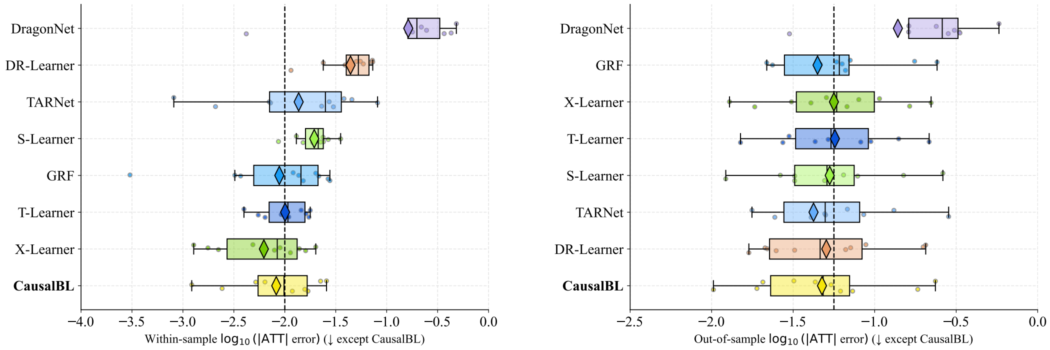

Figure 9: Box plots of absolute ATT error for within-sample (left) and out-of-sample (right) evaluation on the Jobs dataset. Metrics are log-transformed before plotting and methods are ordered by the median error, with CausalBL fixed at the bottom for clarity.

H.4 INTERPRETING BL: A CASE STUDY

H.4.1 INTERPRETING BL(DEEP): HIGH-LEVEL OVERVIEW

Deeper variants of BL are constructed by stacking multiple BL(Single) modules into hierarchical layers, followed by a final affine transformation. This forms a system of interacting UMPs (each of which can be viewed as an agent), where each internal block $\mathcal{B}$ represents a single interpretable UMP. As shown in Figure 10, first-layer modules correspond to individual UMPs, while the second-layer module performs optimal coordination by aggregating or allocating their outputs. This layered structure offers a compositional interpretation of deeper BL models as systems of interacting, interpretable UMPs.

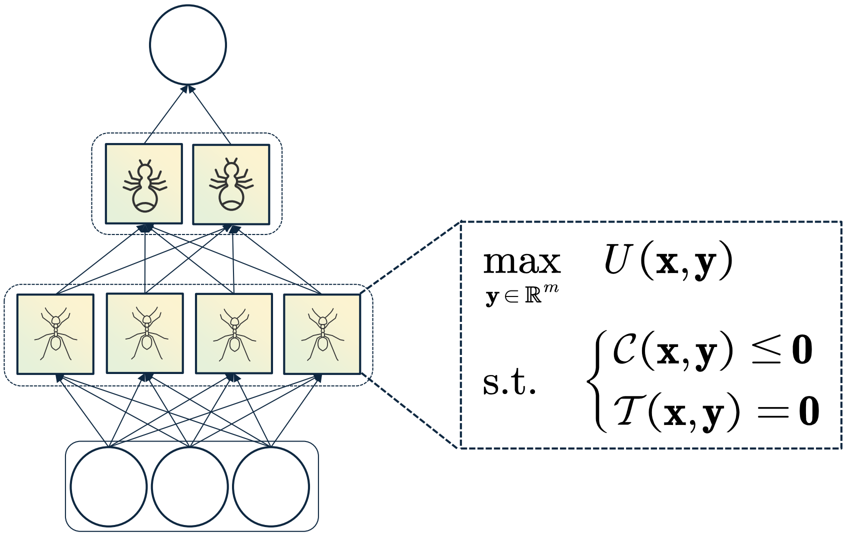
$\max_{\mathbf{y} \in \mathbb{R}^m} \quad U(\mathbf{x}, \mathbf{y})$
s.t. $\begin{cases} \mathcal{C}(\mathbf{x}, \mathbf{y}) \leq \mathbf{0} \\ \mathcal{T}(\mathbf{x}, \mathbf{y}) = \mathbf{0} \end{cases}$

Figure 10: Interpreting deeper BL architectures as hierarchical systems of interacting agents. Each block $\mathcal{B}$ represents an interpretable agent solving its own UMP, while a layer corresponds to a set of heterogeneous agents operating in parallel. The next layer then aggregates and reallocates the negative energies from the previous layer, thereby performing higher-level coordination across agents. This layered organization provides a natural compositional interpretation of deep BL: bottom-layer modules encode local objectives, while upper layers synthesize these into collective outcomes. Analogous structures arise in biological and social systems—for example, in ant colonies, individual ants (first-layer agents) follow simple local rules, yet their collective behavior is coordinated through higher-level interactions (second-layer aggregation), yielding globally efficient resource allocation and task division.

H.4.2 CASE STUDY: ADDITIONAL DETAILS

Table 10: Boston Housing dataset variables and descriptions.

Variable	Description
CRIM	Per-capita crime rate by town
ZN	Proportion of residential land zoned for lots over 25,000 sq.ft.
INDUS	Proportion of non-retail business acres per town
CHAS	Charles River dummy variable (=1 if tract bounds river)
NOX	Nitric oxide concentration (parts per 10 million)
RM	Average number of rooms per dwelling
AGE	Proportion of owner-occupied units built prior to 1940
DIS	Weighted distances to five Boston employment centers
RAD	Index of accessibility to radial highways
TAX	Full-value property-tax rate per $10,000
PTRATIO	Pupil–teacher ratio by town
B	$1000(B_k - 0.63)^2$ where B_k is the proportion of Black residents by town
LSTAT	Percentage of lower-status population
MEDV	Median value of owner-occupied homes in $1000s

Table 11: Semantic roles of blocks in the deep BL architecture.

Layer	Block	Representative preference
Layer 1	Location-Sensitive Buyer	Values river access, transport accessibility, and neighborhood amenities.
	Risk-Sensitive Buyer	Averse to local disamenities such as pollution and environmental risk.
	Economic-Sensitive Buyer	Sensitive to school quality and neighborhood socio-economic composition.
	Zoning-Contrast Buyer	Responds to zoning and land-use patterns that shape local housing supply.
	Affordability-Preferring Buyer	Strongly prefers more affordable housing and dislikes high prices.
Layer 2	Integrated Location–Economic Buyer	Jointly evaluates location and socio-economic attributes in an integrated way.
	Budget-Conflict Buyer	Exhibits strong preferences for desirable locations but faces binding budget constraints.
	Balanced Trade-off Buyer	Jointly considers multiple housing attributes in a balanced manner.
Layer 3	Representative Composite Buyer	Aggregates all lower-level preference components into a representative household.

Table 12: Each block in the deep BL architecture is aligned with a classic preference mechanism documented in the economics literature.

Layer / Block	Representative reference
Layer 1: Location-Sensitive Buyer	Gibbons & Machin (2005)
Layer 1: Risk-Sensitive Buyer	Chay & Greenstone (2005)
Layer 1: Economic-Sensitive Buyer	Black (1999)
Layer 1: Zoning-Contrast Buyer	Glaeser & Gyourko (2002)
Layer 1: Affordability-Preferring Buyer	McFadden (1977)
Layer 2: Integrated Location–Economic Buyer	Bayer et al. (2007)
Layer 2: Budget-Conflict Buyer	Balseiro et al. (2019)
Layer 2: Balanced Trade-off Buyer	Rosen (1974)

H.5 Prediction on High-Dimensional Inputs

Datasets Description and Preprocessing. For image datasets, we use the official train/test splits of MNIST and Fashion-MNIST: Inputs are converted to single-channel images scaled to $[0, 1]$ and standardized with dataset-specific statistics. No resizing or data augmentation is applied. Training uses shuffled mini-batches of size 64. For text datasets, we apply the following procedures:

1 Data sources and official splits. We use the official training and test splits for AG News and Yelp Review Polarity without any custom re-partitioning. Both datasets are class-balanced across labels, and we do not perform any resampling.

2 Dataset sizes. AG News: 120,000 training / 7,600 test samples with four balanced classes. Yelp Review Polarity: 560,000 training / 38,000 test samples with two balanced classes.

3 Label mapping. AG News: labels 1–4 are mapped to 0–3. Yelp Review Polarity: labels 1–2 are mapped to 0–1.

4 Text preprocessing and feature representation. All texts are lowercased and tokenized at the word level. The vocabulary is built with unigrams and bigrams, discarding words that appear fewer than two times in the training corpus. The vocabulary size is capped (AG News: 200,000; Yelp: 100,000). We compute TF–IDF weights on the training split and apply the learned weights to the test split. Dimensionality is reduced to 128 latent components using truncated singular value decomposition (SVD). Features are standardized to zero mean and unit variance and finally ℓ_2-normalized. We fix the random seed for reproducibility and reuse the learned preprocessing components across runs.

Additional OOD Detection Results. In additional to accuracy and AUROC, we also report AUPR and FPR@95 for both image and text datasets; the results are shown in Table 13. On image datasets, BL (depth=1) achieves the best overall balance: it ranks first on Fashion-MNIST AUPR and second on Fashion-MNIST FPR@95. On MNIST, it is second in AUPR but underperforms in FPR@95 compared with E-MLP (depth=2). These results suggest that BL yields separable score distributions, particularly on Fashion-MNIST, although its 95% FPR threshold admits more OOD samples than E-MLP at the same recall. On text tasks, BL remains competitive: BL (depth=3) achieves the lowest FPR@95 on AG News, and BL (depth=2) leads AUPR on Yelp.

Table 13: OOD AUPR and FPR@95 (%) on image and text datasets. BL and E-MLP are evaluated at depths 1–3 with matched parameter counts, both without skip connections. Top-two per column are blue and red.

Model	**MNIST**		**Fashion-MNIST**	
	AUPR	FPR@95	AUPR	FPR@95
E-MLP (depth=1)	89.37 ± 1.52	35.57 ± 5.87	91.35 ± 1.25	28.24 ± 4.37
BL (depth=1)	91.57 ± 2.39	47.81 ± 11.29	91.79 ± 0.90	38.86 ± 2.57
E-MLP (depth=2)	91.52 ± 1.27	28.89 ± 2.85	86.19 ± 2.27	47.72 ± 4.79
BL (depth=2)	91.20 ± 1.22	52.71 ± 18.66	89.30 ± 2.47	42.65 ± 9.53
E-MLP (depth=3)	90.04 ± 1.89	31.92 ± 5.76	84.30 ± 1.50	54.49 ± 2.74
BL (depth=3)	92.36 ± 2.03	32.32 ± 5.76	88.41 ± 4.04	41.19 ± 13.36
Model	**AG News**		**Yelp**	
	AUPR	FPR@95	AUPR	FPR@95
E-MLP (depth=1)	87.06 ± 0.08	91.75 ± 0.15	20.76 ± 0.28	92.27 ± 0.52
BL (depth=1)	89.53 ± 0.05	86.68 ± 0.51	20.49 ± 0.14	97.06 ± 0.05
E-MLP (depth=2)	88.59 ± 0.18	89.26 ± 0.29	20.65 ± 0.54	92.37 ± 1.06
BL (depth=2)	88.06 ± 0.17	86.86 ± 0.79	20.80 ± 0.28	96.95 ± 0.07
E-MLP (depth=3)	89.82 ± 0.35	87.53 ± 0.38	20.74 ± 0.53	92.41 ± 1.00
BL (depth=3)	88.38 ± 0.26	86.04 ± 0.49	20.44 ± 0.19	96.77 ± 0.43

Number of Parameters. To ensure a fair comparison between E-MLP and BL, we match the number of trainable parameters as closely as possible for models with the same depth (see Table 14).

Running Time. In order to further evaluate the computational efficiency of BL, we report the running time comparisons between Energy-based MLP and BL across image and text datasets. Under comparable parameter budgets, we observe that on image datasets BL requires slightly more running time, whereas on text datasets it is considerably more efficient, with running time reduced to roughly one third to one half of that of E-MLP (see Table 15, 16, 17, 18). Importantly, BL achieves better predictive performance while maintaining running time that is similar to standard MLPs, and in some cases even shorter.

Calibration We report ECE and NLL metrics to assess calibration quality, and the results are presented in Table 19. On image datasets, BL provides substantially better calibration, with BL models occupying the top two positions in each column. On text datasets, BL and E-MLP exhibit comparable calibration performance, with no systematic advantage for either model. Overall, these results indicate that BL delivers strong predictive performance together with reliable probability estimates.

Table 14: Number of trainable parameters for E-MLP and BL models across high-dimension datasets.

Dataset	Model	# Parameters
MNIST & FashionMNIST	E-MLP (depth=1)	203,530
	BL (depth=1)	208,384
	E-MLP (depth=2)	235,146
	BL (depth=2)	219,264
	E-MLP (depth=3)	238,314
	BL (depth=3)	221,684
AGNews	E-MLP (depth=1)	136,196
	BL (depth=1)	149,720
	E-MLP (depth=2)	386,284
	BL (depth=2)	397,568
	E-MLP (depth=3)	230,788
	BL (depth=3)	224,128
Yelp	E-MLP (depth=1)	134,146
	BL (depth=1)	148,960
	E-MLP (depth=2)	385,770
	BL (depth=2)	397,312
	E-MLP (depth=3)	230,530
	BL (depth=3)	224,000

Table 15: Comparison of running time between BL and E-MLP on the MNIST dataset

Model	Train Time (s)	Eval Time (s)	Total Time (s)
E-MLP (depth=1)	100.59 ± 0.29	0.33 ± 0.03	100.92 ± 0.30
BL (depth=1)	110.63 ± 3.34	0.23 ± 0.51	110.86 ± 3.85
E-MLP (depth=2)	102.64 ± 0.26	0.35 ± 0.03	102.98 ± 0.26
BL (depth=2)	122.85 ± 3.95	0.28 ± 0.63	123.14 ± 4.58
E-MLP (depth=3)	104.52 ± 0.30	0.32 ± 0.00	104.84 ± 0.30
BL (depth=3)	140.17 ± 4.42	0.00 ± 0.00	140.18 ± 4.43

Table 16: Comparison of running time between BL and E-MLP on the FashionMNIST dataset

Model	**Train Time (s)**	**Eval Time (s)**	**Total Time (s)**
E-MLP (depth=1)	73.57 ± 1.20	0.23 ± 0.01	73.80 ± 1.19
BL (depth=1)	96.52 ± 2.90	0.23 ± 0.51	96.75 ± 3.41
E-MLP (depth=2)	78.25 ± 0.28	0.22 ± 0.01	78.47 ± 0.27
BL (depth=2)	114.43 ± 3.72	0.28 ± 0.63	114.72 ± 4.35
E-MLP (depth=3)	85.57 ± 1.19	0.22 ± 0.01	85.79 ± 1.19
BL (depth=3)	130.03 ± 4.96	0.00 ± 0.00	130.03 ± 4.96

Table 17: Comparison of running time between BL and E-MLP on the AGNews dataset

Model	**Train Time (s)**	**Eval Time (s)**	**Total Time (s)**
E-MLP (depth=1)	60.15 ± 0.47	0.04 ± 0.04	60.19 ± 0.46
BL (depth=1)	22.96 ± 0.79	0.06 ± 0.14	23.02 ± 0.93
E-MLP (depth=2)	65.28 ± 0.32	0.03 ± 0.00	65.31 ± 0.32
BL (depth=2)	28.09 ± 0.79	0.09 ± 0.20	28.18 ± 0.99
E-MLP (depth=3)	68.81 ± 0.28	0.03 ± 0.00	68.84 ± 0.28
BL (depth=3)	34.08 ± 3.55	0.00 ± 0.00	34.08 ± 3.56

Table 18: Comparison of running time between BL and E-MLP on the Yelp dataset

Model	**Train Time (s)**	**Eval Time (s)**	**Total Time (s)**
E-MLP (depth=1)	606.93 ± 3.14	0.23 ± 0.01	607.16 ± 3.14
BL (depth=1)	186.53 ± 1.49	0.02 ± 0.04	186.55 ± 1.51
E-MLP (depth=2)	612.15 ± 6.33	0.34 ± 0.08	612.49 ± 6.32
BL (depth=2)	184.03 ± 1.30	0.00 ± 0.00	184.03 ± 1.30
E-MLP (depth=3)	614.99 ± 6.08	0.27 ± 0.05	615.26 ± 6.06
BL (depth=3)	183.51 ± 1.38	0.00 ± 0.00	183.51 ± 1.38

Table 19: ECE and NLL on image and text datasets. BL and E-MLP are evaluated at depths 1–3 with matched parameter counts. Top-two per column are blue and red.

Model	**MNIST**		**Fashion-MNIST**	
	ECE	NLL	ECE	NLL
E-MLP (depth=1)	0.02 ± 0.00	0.20 ± 0.02	0.08 ± 0.00	0.74 ± 0.01
BL (depth=1)	0.02 ± 0.00	0.26 ± 0.01	0.05 ± 0.00	0.36 ± 0.01
E-MLP (depth=2)	0.02 ± 0.00	0.23 ± 0.02	0.09 ± 0.00	0.89 ± 0.03
BL (depth=2)	0.02 ± 0.00	0.16 ± 0.01	0.07 ± 0.00	0.44 ± 0.01
E-MLP (depth=3)	0.02 ± 0.00	0.16 ± 0.02	0.09 ± 0.00	0.85 ± 0.04
BL (depth=3)	0.02 ± 0.00	0.13 ± 0.02	0.07 ± 0.00	0.49 ± 0.02

Model	**AG News**		**Yelp**	
	ECE	NLL	ECE	NLL
E-MLP (depth=1)	0.01 ± 0.00	0.31 ± 0.00	0.00 ± 0.00	0.22 ± 0.00
BL (depth=1)	0.02 ± 0.00	0.30 ± 0.01	0.01 ± 0.00	0.21 ± 0.00
E-MLP (depth=2)	0.02 ± 0.00	0.30 ± 0.01	0.00 ± 0.00	0.21 ± 0.00
BL (depth=2)	0.02 ± 0.00	0.30 ± 0.00	0.00 ± 0.00	0.21 ± 0.00
E-MLP (depth=3)	0.02 ± 0.00	0.30 ± 0.01	0.01 ± 0.00	0.21 ± 0.00
BL (depth=3)	0.02 ± 0.01	0.32 ± 0.01	0.01 ± 0.00	0.22 ± 0.00

H.6 Evaluating Penalty-Based Constraint Enforcement under Finite Temperature

To evaluate whether the learnable penalty blocks in BL are capable of enforcing near-hard constraints under finite temperature, we isolate the penalty mechanism and test it on a high-dimensional energy-conservation constraint. This diagnostic experiment removes the utility term and focuses solely on the penalty structure, providing a clean characterization of how the penalty block controls constraint violations as a function of temperature τ and penalty scale λ.

Experiment setup. We sample $x \in R^{64}$ i.i.d. from a standard Gaussian $x \sim \mathcal{N}(0, I_{64})$ and define a pure penalty compositional utility

$$T(x,y) \;=\; \|y\|^2 - \|x\|^2, \qquad BL(x,y) \;=\; -\lambda\, T(x,y)^2,$$

which plays the role of an energy-conservation residual and its quadratic penalty.

We target the Gibbs distribution

$$p(y \mid x) \;\propto\; \exp\big(BL(x,y)/\tau\big)$$

using overdamped Langevin dynamics with step size $\eta = 10^{-4}$:

$$y_{k+1} = y_k + \eta\, \nabla_y BL(x, y_k)/\tau + \sqrt{2\eta\tau}\, \xi_k, \qquad \xi_k \sim \mathcal{N}(0, I_{64}).$$

For each pair (λ, τ) we run 512 parallel chains, each for 1500 Langevin steps (500 burn-in). We sweep over temperatures $\tau \in \{2.0, 1.0, 0.5, 0.25, 0.1, 0.05, 0.02, 0.01, 0.005\}$ at a fixed penalty $\lambda = 25$, and over penalty weights $\lambda \in \{0, 1, 3, 10, 30, 100, 200, 500\}$ at a fixed temperature $\tau = 0.05$.

For each configuration we record the residual magnitude $|T(x,y)|$ from the final state of every chain. We then report three summary statistics: (i) the mean violation $E[|T(x,y)|]$, (ii) the 95th percentile of $|T(x,y)|$, and (iii) the empirical probability of near-feasible samples. We declare a sample to satisfy the constraint approximately if

$$|T(x,y)| \;\leq\; \varepsilon_{\text{tol}} \quad \text{with} \quad \varepsilon_{\text{tol}} = 10^{-1},$$

and estimate $P(|T(x,y)| \leq \varepsilon_{\text{tol}})$ across chains. This tolerance scale is chosen to be small relative to the typical unconstrained residuals, so that the near-feasible regime corresponds to a practically tight energy-conservation constraint.

Results. The results (Fig. 11) exhibit the classical behavior of Gibbs-type penalty methods. Both decreasing the temperature τ and increasing the penalty weight λ substantially reduce constraint violation. When τ becomes sufficiently small ($\tau \leq 0.01$), nearly all samples satisfy the constraint threshold. Similarly, increasing λ sharpens the energy well around $T(x,y) = 0$, leading to rapidly diminishing violations. All curves are smooth and monotone in the 64-dimensional setting, indicating excellent numerical stability of the Langevin sampler and demonstrating that the BL penalty structure functions as a genuine, controllable penalty mechanism inside a stochastic energy-based model.

H.7 Parameter Recovery Analysis

We conduct a simulation-based parameter recovery study in a teacher–student setup to examine whether BL can recover the underlying utility and constraint functions. We generate covariates $X \in R^3$ with independent components $x_i \overset{i.i.d.}{\sim} U(-2, 2)$, and define the teacher function

$$Y^\star = x_1 x_2 + 0.5 x_3 + 0.2 x_1 x_3.$$

The observed target variable is generated as

$$Y = Y^\star + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \sigma^2),$$

so that Y represents a noisy observation of the teacher function. A BL student model is then trained on the synthetic pairs (X, Y), and its recovered parameters are compared against the ground-truth teacher parameters to assess recovery performance.

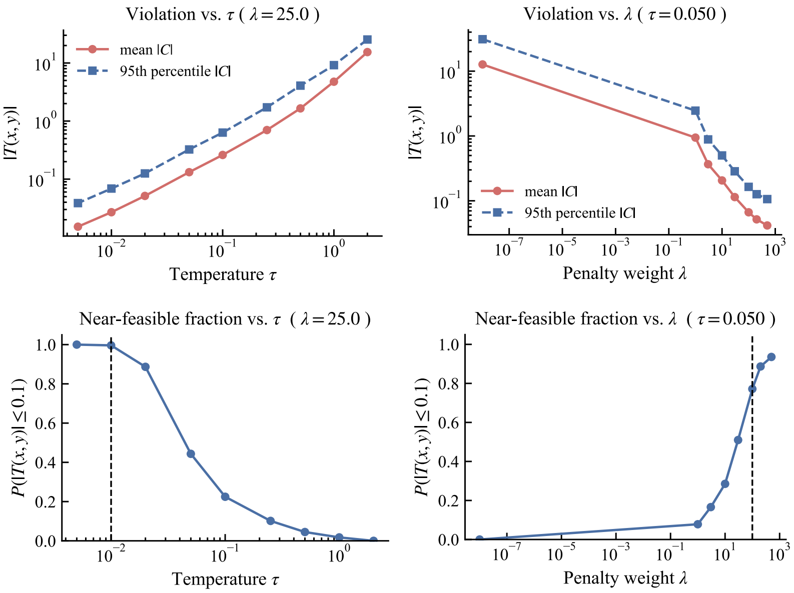

Figure 11: Constraint enforcement test of the BL penalty block on an energy-conservation constraint. The figure reports violation statistics $|T(x, y)|$ when varying the temperature τ (left side of panel) and the penalty weight λ (right side of panel).

- **Depth-1 parameter recovery.** We directly compare the learned parameters with the ground-truth coefficients of the teacher model. As shown in Table 20, the recovered coefficients in the utility block, as well as in both the inequality and equality constraint blocks, are extremely close to the true values, with very small absolute and relative estimation errors.
- **Higher-depth recovery (depth 3 and 5).** At larger depths, individual parameter alignment is less informative because the block compositions become more complex. We therefore assess recovery by comparing the learned and ground-truth block outputs through Q–Q plots. As shown in Figure 12, the points lie almost perfectly on the identity line, indicating that the learned functions accurately reproduce the teacher functions even in deeper architectures.

Table 20: Parameter recovery results for the depth-1 BL model.

Category	Feature	Teacher	Student	Diff	Rel. Error
Utility	x_1	0.38	0.38	0.0030	0.0080
Utility	x_2	0.41	0.41	0.0008	0.0019
Utility	x_3	-0.12	-0.11	0.0025	0.0213
Utility	C	0.45	0.45	0.0069	0.0152
Inequality Constraints	x_1	-0.11	-0.11	0.0025	0.0229
Inequality Constraints	x_2	0.10	0.11	0.0057	0.0553
Inequality Constraints	x_3	-0.25	-0.25	0.0034	0.0140
Inequality Constraints	C	0.29	0.28	0.0069	0.0239
Equality Constraints	x_1	0.44	0.44	0.0022	0.0051
Equality Constraints	x_2	-0.37	-0.36	0.0012	0.0034
Equality Constraints	x_3	0.43	0.44	0.0009	0.0021
Equality Constraints	C	0.09	0.08	0.0064	0.0711

H.8 Empirical Verification of Structural Identifiability

To empirically validate the theoretical identifiability property of the proposed IBL model, we conducted an experiment based on the Jacobian rank criterion. Structural identifiability is ensured when the parameterization yields a full-rank Jacobian almost everywhere (Ljung & Glad, 1994). Follow-

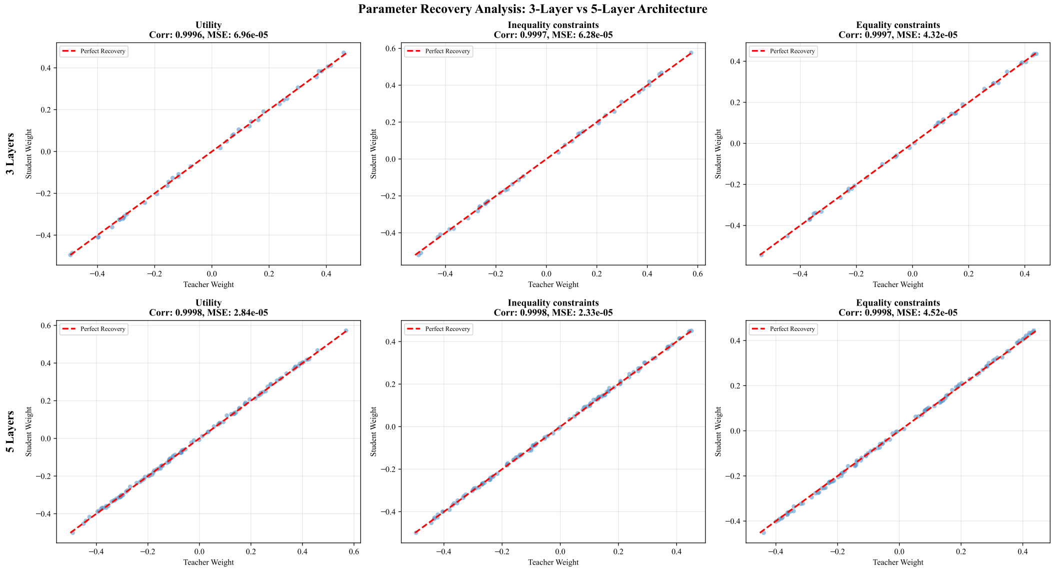

Figure 12: Q–Q recovery plots for 3-layer and 5-layer BL models.

ing this classical criterion, we evaluate whether each model instance maintains a non-degenerate Jacobian under random initialization.

Experimental Setup We evaluate identifiability across three model families: ReLU MLP, Softplus MLP, and IBL. For each architecture depth $L \in \{2, 3, 4, 5, 6, 7\}$, we construct networks with matched parameter budgets and generate 20 random initializations. Each initialized model is assessed by computing the numerical rank of its Jacobian through automatic differentiation. A model is considered identifiable if the Jacobian is full-rank under the singular-value criterion described above. The identifiability ratio for each depth is defined as the proportion of identifiable instances among the total trials.

Results Figure 13 shows that IBL achieves a 100% identifiability ratio across all depths. Softplus MLPs rapidly lose identifiability as depth increases (from 95% at $L = 2$ to 10% at $L = 7$), while ReLU MLPs remain non-identifiable due to degenerate Jacobians. These findings empirically corroborate our theoretical analysis: IBL maintains an identifiable parameterization whereas standard MLPs do not.

H.9 Case Study: Estimation Results of BL on the Boston Housing Dataset

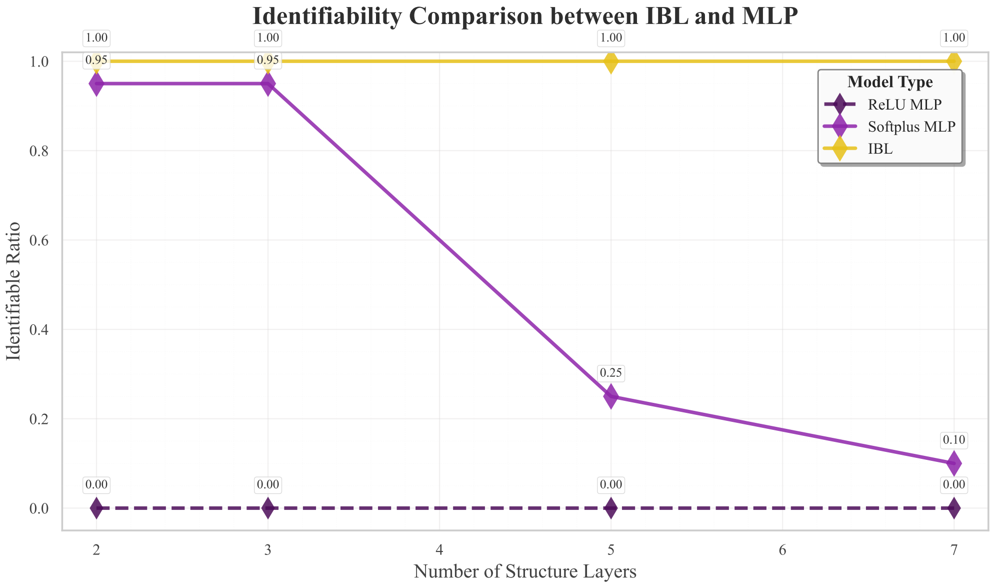

Figure 13: **Identifiability Comparison between IBL and MLP.** The IBL model maintains 100% identifiability across depths, whereas Softplus MLPs suffer from rank degeneracy as layers increase.

Table 21: Estimated UMP block parameters learned by the BL model (layer = [2, 1]) on the Boston Housing dataset. For each block, U denotes the Utility component, C the Inequality-Constraint component, and T the Equality-Constraint component.

	Block 11			Block 12		
Variable	U_{11}	C_{11}	T_{11}	U_{12}	C_{12}	T_{12}
λ	1.003	0.997	0.999	0.997	1.003	1.000
per capita crime rate (CRIM)	0.21	0.14	0.03	0.12	0.09	0.25
residential land proportion (ZN)	0.23	-0.04	-0.27	0.25	0.00	0.09
non-retail business acreage (INDUS)	-0.06	0.21	0.25	0.16	0.22	0.27
Charles River dummy (CHAS)	0.25	0.04	-0.24	-0.12	-0.20	-0.23
nitric oxide concentration (NOX)	-0.06	-0.13	0.21	0.16	0.02	-0.28
average rooms per dwelling (RM)	0.06	0.07	0.05	0.05	-0.19	-0.22
proportion of older units (AGE)	-0.13	-0.12	-0.09	0.14	0.08	-0.18
distance to employment centres (DIS)	0.16	-0.03	0.17	-0.17	-0.09	0.11
radial highway accessibility (RAD)	0.24	-0.11	0.04	-0.28	0.09	0.10
property tax rate (TAX)	-0.20	0.18	0.22	-0.11	-0.06	0.23
low-income population (LSTAT)	0.05	-0.12	-0.09	0.23	-0.16	-0.19
median home value (MEDV)	0.21	-0.08	0.07	0.08	-0.17	0.15
Constant term (C)	0.03	-0.17	-0.07	0.11	-0.16	-0.12

	Block 21		
Variable	U_{21}	C_{21}	T_{21}
λ	1.000	1.003	0.999
Block 11 output ($b_{1,1}$)	0.428	-0.551	0.147
Block 12 output ($b_{1,2}$)	-0.168	-0.356	-0.178
Constant term (C)	0.406	0.219	0.421

Table 22: Estimated UMP parameters for the Layer 1 blocks of the BL model (layer = [5, 3, 1]) trained on the Boston Housing dataset. Here, U denotes the Utility component, C the Inequality-Constraint component, and T the Equality-Constraint component.

Variable	U_{11}	U_{12}	U_{13}	U_{14}	U_{15}
λ	1.000	0.998	1.003	1.002	1.000
per capita crime rate (CRIM)	0.21	0.12	0.17	-0.09	0.06
residential land proportion (ZN)	0.23	0.25	-0.07	-0.22	-0.16
non-retail business acreage (INDUS)	-0.06	0.16	0.16	0.23	-0.14
Charles River dummy (CHAS)	0.25	-0.12	-0.22	-0.05	-0.01
nitric oxide concentration (NOX)	-0.06	0.16	-0.14	0.24	0.16
average rooms per dwelling (RM)	0.05	0.05	0.08	0.09	-0.07
proportion of older units (AGE)	-0.13	0.14	0.06	-0.23	-0.15
distance to employment centres (DIS)	0.16	-0.17	-0.07	0.19	-0.10
radial highway accessibility (RAD)	0.24	-0.27	0.17	-0.08	-0.20
property tax rate (TAX)	-0.20	-0.11	0.19	-0.11	0.10
low-income population (LSTAT)	0.05	0.23	-0.15	-0.28	-0.26
median home value (MEDV)	0.21	0.08	0.25	0.08	0.06
Constant term (C)	0.03	0.12	-0.09	-0.06	0.15
	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}
λ	0.999	1.001	1.000	0.997	1.002
per capita crime rate (CRIM)	0.13	0.09	-0.10	0.11	0.05
residential land proportion (ZN)	-0.04	-0.01	-0.27	-0.23	-0.10
non-retail business acreage (INDUS)	0.21	0.22	-0.16	0.20	0.15
Charles River dummy (CHAS)	0.04	-0.19	0.07	-0.20	0.15
nitric oxide concentration (NOX)	-0.13	0.02	-0.04	-0.05	0.11
average rooms per dwelling (RM)	0.07	-0.19	-0.20	0.06	-0.05
proportion of older units (AGE)	-0.13	0.08	0.01	0.14	-0.07
distance to employment centres (DIS)	-0.03	-0.09	-0.19	0.22	0.03
radial highway accessibility (RAD)	-0.11	0.08	-0.23	0.25	-0.05
property tax rate (TAX)	0.18	-0.06	-0.15	-0.22	-0.08
low-income population (LSTAT)	-0.13	-0.17	-0.18	-0.12	0.24
median home value (MEDV)	-0.08	-0.17	0.28	-0.03	-0.03
Constant term (C)	-0.17	-0.16	0.05	-0.21	-0.06
	T_{11}	T_{12}	T_{13}	T_{14}	T_{15}
λ	0.999	1.002	0.999	1.004	1.001
per capita crime rate (CRIM)	0.03	0.25	0.08	0.25	0.00
residential land proportion (ZN)	-0.27	0.10	-0.26	-0.20	-0.02
non-retail business acreage (INDUS)	0.25	0.26	-0.18	0.15	0.07
Charles River dummy (CHAS)	-0.23	-0.23	-0.09	0.10	0.08
nitric oxide concentration (NOX)	0.21	-0.28	0.04	0.09	-0.25
average rooms per dwelling (RM)	0.05	-0.21	-0.24	-0.15	-0.10
proportion of older units (AGE)	-0.09	-0.19	-0.12	0.26	0.24
distance to employment centres (DIS)	0.17	0.12	-0.17	0.06	0.10
radial highway accessibility (RAD)	0.04	0.10	0.00	0.04	-0.01
property tax rate (TAX)	0.22	0.23	-0.10	-0.24	-0.17
low-income population (LSTAT)	-0.09	-0.19	-0.19	-0.04	-0.25
median home value (MEDV)	0.08	0.15	-0.19	-0.13	-0.09
Constant term (C)	-0.07	-0.11	-0.16	0.24	0.10

Table 23: Layer 2 and Layer 3 UMP parameters (U, C, T) for Blocks in the BL model (layer = [5, 3, 1]).

Variable	Block 21			Block 22			Block 23		
	U_{21}	C_{21}	T_{21}	U_{22}	C_{22}	T_{22}	U_{23}	C_{23}	T_{23}
λ	1.000	1.000	1.000	0.999	1.003	1.002	1.001	1.002	0.999
Block 11 output ($b_{1,1}$)	0.28	0.06	-0.20	-0.31	0.24	0.18	-0.29	-0.08	0.22
Block 12 output ($b_{1,2}$)	0.21	-0.11	-0.09	-0.44	0.12	-0.22	0.15	-0.22	0.20
Block 13 output ($b_{1,3}$)	-0.40	0.18	-0.44	-0.36	-0.01	-0.09	-0.13	-0.14	0.32
Block 14 output ($b_{1,4}$)	-0.27	-0.17	0.30	0.33	-0.34	-0.26	0.28	-0.42	-0.34
Block 15 output ($b_{1,5}$)	-0.07	-0.29	0.34	0.22	-0.38	-0.08	-0.14	0.25	0.32
Constant term (C)	0.43	0.33	0.16	0.38	-0.42	-0.32	-0.33	-0.31	-0.21

Variable	U_{31}	C_{31}	T_{31}
λ	1.002	0.998	1.000
Block 21 output ($b_{2,1}$)	0.21	-0.13	0.36
Block 22 output ($b_{2,2}$)	0.54	-0.48	0.43
Block 23 output ($b_{2,3}$)	-0.08	0.28	0.55
Constant term (C)	-0.01	-0.58	-0.14